GE 120/GE 130

ETOS Programmer's Manual

GE 100 LINE

BULL

GENERAL ELECTRIC

Code 3918106 B

Réf. 32.38.001 A

Code 3918106 B

Réf. 32.38.001 A

GE 120/GE 130
ETOS Programmer's Manual

AUGUST 1970

BULL
GENERAL ELECTRIC

PREFACE

This manual is produced by General Electric Information Systems Italia. The terms used conform to the standard IFIP/ICC terminology.

Suggestions and comments pertaining to form, content, purpose, or use of the manual are invited. The Document Review sheet in the back of the book is provided for this purpose.

(C) 1970 by General Electric Information Systems Italia

TABLE OF CONTENTS

	PART	PAGE
PREFACE		
INTRODUCTION		ix
I. GENERAL DESCRIPTION	I	1
A. WHAT IS ETOS	I	1
1. Definition	I	1
2. Components	I	2
3. Related Software	I	3
4. Major Block Diagram	I	4
B. HOW TO USE ETOS	1	6
1. General Concepts	I	6
2. Applications	I	8
3. Hardware Configuration	I	10
4. Core Store Allocations	I	10
II. TAPE AND PROGRAM ORGANIZATION	II	1
1. General Concepts	II	1
2. System Tape (ST)	II	2
3. Library Tape (LT)	II	8
4. Master Tape (MT)	II	9

	PART	PAGE
III. ETOS OPERATION	III	1
A. USER CONTROL OF ETOS	III	1
1. Parameter Cards for Order Programs	III	1
2. Pilot 0	III	7
B. SUPERVISOR	III	7
1. Purpose and Organization	III	8
2. Pilot 1	III	10
3. Pilot 2	III	12
4. Segment Load Routine	III	13
5. Execution of Order Programs	III	14
6. Basic Input/Output System (BIOS)	III	14
7. System Tape Mode (STM)	III	16
8. Master Tape Mode (MTM)	III	16
a. General Concepts	III	16
b. Monitor	III	17
9. Supervisor Condition Checking	III	19
C. INDIVIDUAL ORDER PROGRAMS	III	22
1. "INITIALIZATION" Orders	III	24
a. .INT Initialize	III	24
b. .TLW Tape Label Write	III	29

	PART	PAGE
2. "TRANSLATE and GO" Orders	III	34
a. .APS Assemble APS Program	III	34
b. .COB Compile COBOL Program	III	36
c. .FOR Compile FORTRAN Program	III	37
d. .SPG Generate Sort/Merge Program	III	39
e. .API Assembled Program Insert	III	41
f. .PRL Program Load	III	45
g. .APP Assembled Program Punch	III	53
h. .SBP Subroutine Punch	III	56.1
3. "DEBUGGING" Orders	III	57
a. .TDC Tape Duplicate	III	57
b. .DPG Debugging Program Generate	III	63
c. .SGE Samples Generate	III	70
d. .TDP Tape Dump	III	78
e. .CRD Card Dump	III	82
f. .MDP Memory Dump	III	84
g. .PCC Punched Card Check	III	85.0
h. .PSC Punched Subroutine Check	III	85.4
i. .PSI Punched Subroutine Insert	III	85.5
j. .TCK System Tape Checking	III	85.15
4. "PROGRAM TAPE GENERATE AND MAINTAIN" Orders	III	86
a. .STG System Tape Generate	III	86
b. .SPD System Tape Print and Duplicate	III	94
c. .STC System Tape Close	III	97
d. .STU System Tape Update	III	99
e. .STS System Tape Strip	III	107
f. .SLS Subroutine Library Strip	III	111
g. .SLI Subroutine Library Insert	III	115
h. .SPC Subroutine Peripheral Change	III	122
i. .LTU Library Tape Update	III	126

			PART	PAGE
j.	.MTG	Master Tape Generate	III	135
k.	.BPR	Basic Program Replace	III	146
l.	.LPD	Library Tape Print and Duplicate	III	150.1
m.	.LTM	Library Tape Merge	III	150.4
n.	.TPU	ST Program Update	III	150.8
5.	"OTHER" Orders		III	151
a.	.CMT	Comment	III	151
b.	.HLT	Halt	III	152
c.	.MTS	Master Tape Stop	III	153
d.	.RTS	Restart	III	154
e.	.CDC	Cards Duplicate	III	159
IV.	EXAMPLES		IV	1
A.	ASSEMBLE AND DEBUG APS PROGRAM		IV	1
B.	INSERT UPDATED S/R INTO ST		IV	4
V.	APPENDIXES		V	1
A.	GLOSSARY OF TERMS		V	1
B.	PREREQUISITE AND RELATED READING		V	7
C.	SYSTEM RESERVED AREA (SRA) CHART		V	8
D.	FILE TABLES AND LBC CARDS		V	14
E.	SYSTEM TAPE GENERATION		V	21
F.	TAPE UNITS REQUIRED FOR INDIVIDUAL ORDERS		V	26
INDEX OF TERMS			Y	1

LIST OF ILLUSTRATIONS

FIGURE	TITLE	PART	PAGE
1	ETOS Major Block Diagram	I	5
2	Organization of ST and MT	II	2
3	Types 1 and 3 Program Label Format	II	3
4	Types 1 and 3 Program Block Label Format	II	4
5	Type 2 Program Library Format	II	6
6	Type 2 Program Block Label Format	II	6
7	Organization of LT	II	9
8	Arrangement of Program on MT		
9	Format of Monitor	II	11
10	Basic Format of Order Cards	III	2
11	Format of Supervisor on ST and MT	III	8
12	Order Program Flowchart	III	11
13	APS Program Assembly/Debug Flow chart	IV	2
14	Replace Subroutine Flowchart	IV	5

INTRODUCTION

This manual describes ETOS (Extended Tape Operating System) the software environment provided for GE-120 and GE-130 Magnetic Tape oriented systems. Since this manual was originally developed for the GE-130, any reference to this system should be extended to the GE-120, except when some differences between the two systems are brought out. The material is presented in five parts, as follows:

I. General Description

II. Tape and Program Organization
This part describes the formats of the programs recorded on the three Program Tapes used by ETOS.

III. ETOS Operation
This part describes the components of ETOS (Supervisor and Order Programs) and the Parameter Cards prepared by the user to direct the operation of ETOS.

IV. Examples

V. Appendixes

The manual is written for use by Programmers, Analysts and Applications Engineers. A seperate manual, "ETOS Operator's Manual" is provided for the use of Operators.

A comprehensive listing of prerequisite and related reading will be found in Appendix B. The reader will find this listing of value in relating this manual to the GE-120 and GE-130 system concepts.

PART I - GENERAL DESCRIPTION

A. WHAT IS ETOS

1. Definition

ETOS is a comprehensive set of programs providing a software environment in which GE-120 and GE-130 tape-oriented systems can operate with maximum efficiency and reliability, and with minimum operator intervention. Basically it is used in two ways.

- To translate source programs into machine language and debug the resultant object programs. This series of operations is executed in the System's starting phase and is repeated each time new applications are programmed, or where modifications are made to existing applications.
- To control the execution of user programs.

ETOS offers many advantages to the user, including:

- Automatic link-up between the different work phases, with minimum operator intervention and consequent errors. It is possible, for example, to compile a program, generate sample files, start the compiled program; and execute auxiliary operations such as file or store dumping, with no operator intervention when passing from phase to phase. This makes it possible to execute a whole series of programs without interruption, by first preparing the link-ups between the various programs.
- Standardization of communications between programs and operator. For example, all programs print standard messages and halt at fixed addresses if errors or incidents should occur. This simplifies the operator's work considerably.

- Independence from different models of main peripheral units (printers, card readers, etc.) which may be connected to the system. All I/O (Input/Output) routines are stored in fixed locations and are supplied in different versions according to system configurations. As a result, a print, read, or write instruction normally will obtain the same result, regardless of the model of the peripheral unit used.

- The program designed for the GE-115 TOS (Tape Operating System) may be run unchanged on the GE-120 or GE-130, under control of ETOS. This is done either by re-translating them, or by inserting them in machine language into GE-120 or GE-130 program tapes.

2. Components

ETOS programs can be divided into two main groups:

a. The Supervisor. Located in Core Store during operation, it contains the following programs which are prerequisite for running any other system program:

   - Pilot 1 and Pilot 2 programs
   - BIOS (Basic Input/Output System)
   - Segment Load routine

   The Supervisor is described in Part III, starting on page 7.

b. The Order Programs are selected by the user to direct the actual operation of the GE-130, from program translation and debugging to the execution of user programs and maintenance of the software library. The Order Programs are described in Part III, starting on page 22.

3. Related Software

In addition to user programs, a number of general purpose programs are used, including:

- Translator Programs
- Utility Programs
- Standard and User Subroutines (S/R's)

a. Translator Programs

These include assemblers, compilers, and program generators which are used for converting or translating from source to object programs.

- APS (Assembly Programming System) is a symbolic, machine-oriented source language with essentially a one-to-one correspondence between source and machine (object) language. However, it also includes a group of macro-instructions for calling subroutines and EIOS (Extended Input/Output System). APS is fully described in the GE-120 and GE-130 Reference Manuals and the GE-120/130 APS Manual, both of which are listed in Appendix B of this book.

- COBOL (Common Business Oriented Language) is a symbolic, user-oriented language designed for accounting and administrative applications. Each COBOL program can be executed in computer systems that are different in their internal logic and machine language. COBOL is fully described in the COBOL Reference Manual, listed in Appendix B of this book.

- FORTRAN (Formula Translator) is a symbolic, user-oriented language designed for mathematical, technical and scientific applications. Each FORTRAN program can be executed in computer systems that are different in their internal logic and machine language. This language is fully described in the FORTRAN Reference Manual, listed in Appendix B of this book.

- SPG (Sort Program Generator) produces optimal Sort and Merge program using simple, user-written SPG statements. This language is especially designed to describe data files to be sorted and/or merged. It is fully described in the SPG Reference Manual, listed in Appendix B of this book.

b. Utility Programs

These programs are designed to perform media conversion operations.

c. Standard and User Subroutines

Standard Subroutine include:

- EIOS. A set of S/R's designed for logical processing of files (as contrasted with physical processing performed by BIOS) on all types of input/output media. APS macro-instructions are used to call EIOS and decribes the data files to be processed. See the GE-120/130 APS Manual for a full description of EIOS.

- Additional S/R's for handling peripheral units that are not covered by EIOS or BIOS.

User S/R's include an extensive S/R library developed by the manufacturer plus those developed by the user for his special needs.

4. Major Block Diagram

Figure 1 is designed to give a visual concept of the components of ETOS, and its related software. It is intended to illustrate the inter-relationships between functions and to show the general operational sequences of the system.

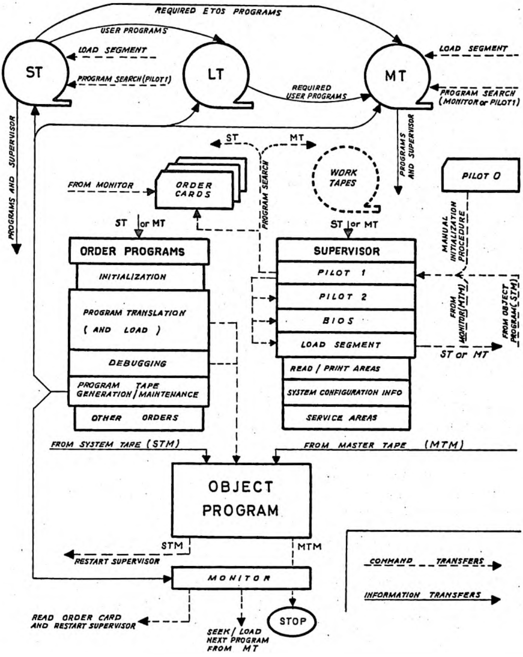


FIGURE 1 – ETOS MAJOR BLOCK DIAGRAM

B. HOW TO USE ETOS

1. General Concepts

ETOS operation is based on the use of three Program Tapes:

System Tape	(ST)
Library Tape	(LT)
Master Tape	(MT)

One, two or three Work Tapes (WT), and the Order cards

a. The ST

This is the basic Program Tape, containing the complete software system. The LT and MT are derived from it, either directly or indirectly. It is made up of:

- Supervisor
- Order Programs
- Translator Programs
- Subroutine Library
- Utility Programs
- User Programs

The ST is used to compile and debug User Programs. When a program has been debugged it is normally transferred to the LT, but it may be started from the ST itself.

b. The LT

This tape contains all the User (and Utility) programs in working order.. However, since it does not contain the ETOS program, it cannot be used for executing user programs.

c. The MT

This tape contains all user and ETOS programs needed for operations during a specific working period (shift, day, week). The programs contained in this tape have been debugged and are arranged in proper sequence for carrying out the planned schedule of operations. They include:

- Required User (and Utility) programs
  (Those needed for the work period)
- Supervisor
- ETOS programs (for initializing the system, loading and starting the user programs)
- Monitor programs (used for chaining each two user programs)

However, additional ETOS programs may be inserted into the MT as desired by the user.

The MT is generated by combining the required ETOS programs from the ST, with the required user (and utility) programs from the LT.

d. The WT

At least two Work Tapes (WT1 and WT2) are required in ETOS operation for storing data or program files. An optional WT3 may be included, as needed.

e. Order Cards

These are used to command the execution of desired ETOS programs, which are thus called "Order Programs". The cards are punched and sequenced by the user as desired, to obtain specific results from the ETOS programs.

2. Applications

ETOS programs may be called up by the Order cards in any or all of the functional phases of the GE-120/130 systems, as follows:

- Program Translation
- Program Debugging
- Program Storage and Maintenance
- Program Execution

a. Program Translation

APS, COBOL and FORTRAN programs are translated into machine language in a single processing run. SPG requires two runs:

- To obtain the program in APS
- To translate the program into machine language

The two runs are linked automatically, without operator intervention.

In the translation process, S/R's can be assembled to the main user program. S/R calling is implicit in some source language statements, or may be carried out by the "SUB" statement.

b. Program Debugging

ETOS provides a number of very helpful facilities for program debugging. For example:

- Recording sample files on tape before user program start-up.

- Presetting user programs for execution with "Path Trace Printing" (printing out all jump instructions that have actually occurred).
- Presetting user programs for "controlled execution". This is done by inserting machine language corrections from punched cards into the translated program, and by presetting some points in the program where printout of significant Core Store areas is desired (Memory Trace Printing).
- Interrupting the debugging at any desired point to perform various checks such as "Tape Dump" and "Memory Dump".

c. Program Storage and Maintenance

After debugging, the program is corrected (if necessary) in symbolic language, retranslated, and finally inserted in an appropriate file with all other debugged programs of the system. This file, the Library Tape (LT), is used as a "file collector" of completed, debugged, user programs.

d. Program Execution

In normal system operation ETOS provides many aids. After the programs to be executed have been selected and arranged in proper sequence, link-up procedures between programs (the Monitors) are prepared, and operator intervention points are established.

When these decisions have been made by the programmer, the programs are collected by ETOS in a specific object-program tape (the Master Tape, or MT) and the Monitors are inserted. Depending on the user's initial decisions, the Monitor automatically loads and starts the next program. The operator intervenes only at pre-established points: to mount new tapes, change printer paper, or to modify the order of execution of the program.

3. Hardware Configurations

ETOS requires the following minimum system configuration:

- 16K Core Store
- Any one Card Reader or Card Reader/Punch
- Any one Printer
- Four Magnetic Tape Units

4. Core Store Allocations

In using ETOS, Core Store is divided into two areas:

- System Reserved Area (SRA)
- User Reserved Area (URA)

The SRA requires approximately 2400 octets of storage and includes:

- Supervisor
- Data required by user programs.
- Data required by Order programs.
- HLT instructions (where programs stop as a result of condition checking).
- Instructions for calling EIOS and other standard routines.

SRA is described in detail in Appendix C.

The URA is available for user programs and will vary according to the total Core Store capacity.

PART II - TAPE AND PROGRAM ORGANIZATION

1. GENERAL CONCEPTS

This part deals with the organization and formats of programs on the ST, LT and MT.

NOTES:

a. ST, MT, and LT tapes are generated by the appropriate Order programs.

b. The three tapes may be either 7- or 9-track.

c. Only characters of the GE-100 character set are recorded.

d. There are three types of programs, as follows:

Type	Description
1	ETOS Programs and Translators
2	Subroutines
3	User and Utility Programs

e. Many terms used in the following descriptions are abbreviated for the sake of simplification. If the reader wishes to refresh his memory, refer to Appendix A, "Glossary of Terms", or to the Index of Terms.

2. SYSTEM TAPE (ST)

The organization of programs on the ST can be visualized as follows:

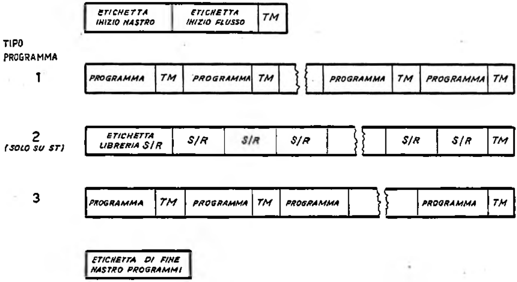


FIGURA 2 – ORGANIZZAZIONE DEL ST E DEL MT

NOTES: 1) Program Tape Header Labels are the same as for standard Data Tapes, and are identified as follows:
VOL1 for 9-track tape labels, or
INT1 for 7-track tape labels
HRD1 for file labels

2) The Program Tape Trailer Label consists of 28 characters, of which the first nine are exclamation marks (!).

Type 1 and Type 3 programs have similar formats, and are described together. Type 2 programs are described separately.

a. Type 1 and Type 3 Programs

The label of each program is 28 characters long, formatted as follows:

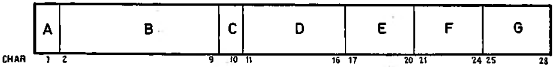


FIGURE 3 - TYPES 1 AND 3 PROGRAM LABEL FORMAT

Field	ETOS and Translator Programs	User and Utility Programs	Notes
A	1	3	Program Type
B	Program Name	Same as Type 1	Alphameric
C	O, if GE-115 BIOS, no tape routine 1, if GE-115 BIOS, with tape routine 4, if GE-130 BIOS, no tape routine 5, if GE-130 BIOS, with tape routine	C, if program generated by .DPG order and uses GE-130 BIOS. Otherwise, same as for Type 1 programs	Indicate type of BIOS

Field	ETOS and Translator Programs	User and Utility Programs	Notes
D	ƀYYDDD	ƀYYDDD	Date
E	Not significant	Starting address (in unpacked binary) of File Table in Core Store, if any. Otherwise, all zeros.	
F	Program starting address (in unpacked binary)	Same as Type 1	
G	Address (in unpacked binary) of first store location not used by program	Same and Type 1	

Each program contains one or more segments, each of which contains one or more blocks. Each block has a label of 20 characters, formatted as follows:

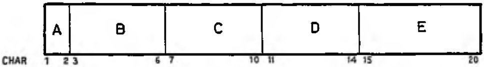


FIGURE 4 - TYPES 1 AND 3 PROGRAM BLOCK LABEL FORMAT

Field	Contents
A	??, if block is the last in segment. 00, if not the last.
B	Number (in unpacked decimal) of segment to which block belongs.
C	Block length, minus 1 (in unpacked binary).
D	Initial address of block in Core Store (in unpacked binary).
E	Not significant.

NOTES:

1) The segments within a program are numbered sequentially from 01 to 98 (Field B).

2) Block lengths have the following limitations:
   minimum - 20 characters
   maximum - 4096 characters

3) The Supervisor is the first Type 1 program. Its format differs from other programs in that:
   - It uses no block labels
   - It is always eight blocks long.

   The program name (Field B, figure 3, Page II-3) for the Supervisor is BASICPRG.

4) In Type 3 programs the File Table, if present, occupies the first block of the first segment. The last character in the File Table is an exclamation mark(!).

5) Type 3 programs generated by the .DPG Order (for controlled execution) are always recorded after the normal Type 3 programs.

b. Type 2 Programs (S/R Library)

Figures 5 and 6 illustrate the structure of the Type 2 programs.

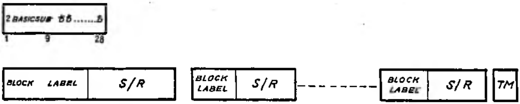


FIGURE 5 - TYPE 2 PROGRAM LIBRARY FORMAT

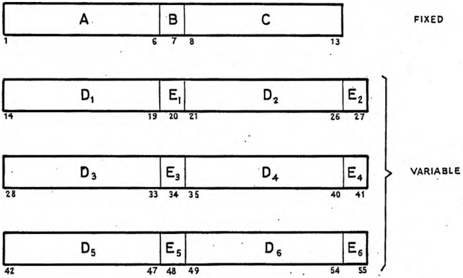


FIGURE 6 - TYPE 2 PROGRAM BLOCK LABEL FORMAT

Field	Contents
A	Subroutine (S/R) Name.
B	Number of S/R's called (maximum 6).
C	Date S/R inserted in Library (ƀYYDDD).
D 1-6	Name of S/R called, if any.
E 1-6	Sequence number of S/R called, if any.

NOTES:

1) The S/R Library label uses only fields A and B of the Type 1 and Type 3 program labels on ST (see figure 3, page II-3) as follows:
   Field A = 2 (type of program)
   Field B = BASICSUB (S/R Library Name)
   Fields C through G are blank characters.

2) Each S/R may occupy one or more blocks.

3) Each block in the S/R Library consists of:
   - Block label.
   - Four characters, containing length (in unpacked binary) of the S/R portion (assumed to be in packed form) contained in the block.
   - Four characters containing initial address (in unpacked binary) of block in Core Store.
   - Part or all of a S/R.

4) Block lengths may be up to 1200 characters.

5) Each block is followed by either:

   E - If block is the last of the S/R, or
   R - If block is not the last of the S/R

6) S/R's are not preceded by labels since the identification is contained in each block label.

7) Each S/R may call up to six S/R's, numbered 1 to 6 (Field B, figure 6, page II-6).

8) The S/R sequence number (Field E, figure 6) is used to call the desired S/R.

9) Block label lengths:
   Minimum (No S/R's called) = 13 characters
   Maximum (6 S/R's called) = 55 characters

10) S/R's are recorded in machine language and start from address zero. S/R addresses are converted into actual Core Store addresses at assembly time.

3. LIBRARY TAPE (LT)

The LT contains the Type 3 (User and Utility) program in working order; that is, those that have been properly debugged and with all related S/R's incorporated. The organization of programs on the LT can be visualized as follows:

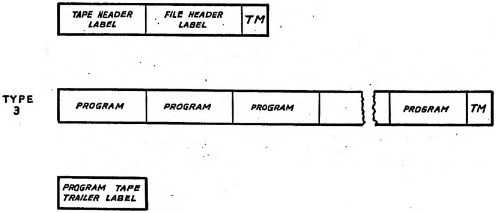


FIGURE 7 - ORGANIZATION OF LT

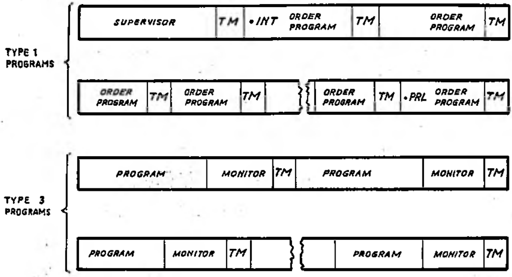
NOTES:

1) The organization and format of programs on the LT is the same as on the ST (figure 2, 3, and 4) except that there are no Tape Mark (TM) separations between each two programs.

2) Programs for controlled execution (generated by .DPG Order program) are not transferred from ST to LT.

4. MASTER TAPE (MT)

The MT contains all Type 3 programs in proper sequence and ready for execution during a user-defined work schedule, plus the Type 1 programs required to execute them. The MT format is the same as the ST format (figures 2, 3, and 4). The following figure illustrates the arrangement of Type 1 and Type 3 programs in the MT.

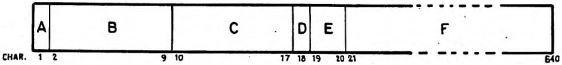


NOTES: 1) The Supervisor, the .INT and .PRL Order programs are inserted automatically in the MT when it is generated, and in the order shown.

2) Any additional Type 1 program may be inserted as needed by the user, before the .PRL Order program.

Rev. Nov. 1969

3) Segments 00 and 99 are automatically inserted before and after each Type 3 program, during MT Generation. Segment 00 contains the appropriate BIOS routines for that program and some optional routines to preset the program for execution. The Monitor is in segment 99. It is used to load and start the next program to be executed. Its format is shown in figure 9.



Field	Contents
A	Monitor Routing Code (may be 1 through 5). Note: the uses of Routing Code are described on Page III, 18
B	Name of program to be loaded: if Field A = 1, or if Field A = 2 and Switch 2 is off.
C	Name of program to be loaded: if Field A = 2 and Switch 2 is on.
D	Not significant

Field	Contents
E	Address relative to beginning of the following list (Field F) where name of current program is found.
F	List of names of Type 3 programs preceding this Monitor.

PART III - ETOS OPERATION

The following pages describe the ETOS programs, and how the user controls them.

ETOS operation is under the control of either the ST or the MT, as desired by the user. Thus, two modes of operation may be defined:

- System Tape Mode (STM)
- Master Tape Mode (MTM)

The differences between the two modes are due to variations in Type 1 programs; principally in the Supervisor. Therefore, the description of the Supervisor includes a discussion of the STM and MTM (see page III-16)

A. USER CONTROL OF ETOS

The system is initialized by a punched card containing the program, Pilot 0. The operation of the system is controlled by Order programs. The parameters of the Order programs are determined by the user, according to his specific needs.

1. Parameter Cards for Order Programs

This chapter gives a general description of Parameter cards, their purpose and applications. How they are processed by the Supervisor is described in Part III, starting on page 10. The types of Parameter cards (Order, Detail, and End) are described in the following paragraphs. Their uses in individual Order programs is described in Part III, starting on page 22.

To obtain service from ETOS the user formulates suitable commands that are communicated to the system by selected Parameter cards. For example, if an APS program on cards is to be assembled, the command is transmitted by an Order card, followed by the source program deck.

After ETOS has executed a given command it seeks another Order card. Thus it can execute a series of programs without interruption, allowing the user to plan his work schedule by preparing a series of Parameter cards.

The Parameter cards for each Order program are used in the following sequence:

One Order card, always present.
One or more Detail cards, where applicable.
One End card, where applicable.

a. Order Cards

The general format of these cards is illustrated as follows:

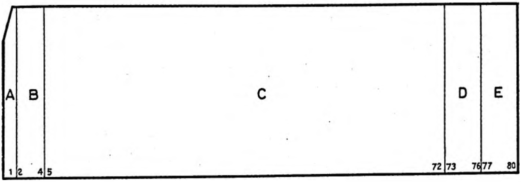


FIGURE 10 - BASIC FORMAT OF ORDER CARDS

Field	Columns	Contents
A	1	The fixed character "." (period).
B	2 - 4	Order name, expressed in mnemonic form.
C	5 - 72	Order information. Contents will vary depending on individual Order.
D	73 – 76	ID Sequence Code if applicable.
E	77 - 80	Always blank.

b. Detail Cards

The first four columns are punched b̸XXX. The symbol "b̸" represents a blank and "XXX" represents the mnemonic name of the card. The rest of the card is formatted in accordance with the specific Order.

c. End-of-Order Cards (End Cards)

These always contain a card name (b̸END) in the first four columns. The rest of the card is either blank or formatted in accordance with the Specific Order. End cards are always required when Detail Cards are present, or when optional but not used.

d. ID Sequence Code

This is commonly referred to as the "ID Code" (Field D, figure 10). When present, it is used to identify a sequence of Order cards which are being used to perform a series of inter-related operations.

When used, the ID Code provides the following checks:

1) An Order card is processed by ETOS, only if the previous Order card in the same sequence was successfully processed. Any intermediate Order cards with no ID Code do not interrupt the sequence.

2) If an Order card is initiating a new sequence, or if it has no ID Code, it is processed regardless of the correct or incorrect execution of the previous Order.

NOTE: The ID Sequence Validity Switch is tested for the correct or incorrect execution of a program. This is an octet in SRA which sets to:
'YES' (hexadecimal 00) if the program was correctly executed.
'NO' (hexadecimal 01) if an error occurs.
The ID Switch can be set by both Type 1 and 3 programs.

e. Examples of Use of Parameter Cards

Problem 1

A program written in APS is to be translated, loaded and started. Each Order to be executed only if previous one was carried out successfully.

Solution: Use following sequence of Parameter cards, with same ID Code (AAA1).

No.	Operation Description	Parameter Cards	ID Code
1	Translate APS program	.APS Order card	AAA1
2	Introduce APS source program.	(source program deck)	

3	Insert assembled program in ST	.API Order card	AAA1
4	Generate Sample File	.SGE Order card	AAA1
5	Introduce information to be recorded on tape	Detail cards	
	End of .SGE Order	End card	
6	Load and start program	.PRL Order card	AAA1
7	Update File Table, if applicable	LBC Detail cards	
	End of .PRL Order	End card	

Problem 2

Same as Problem 1, plus print the file recorded by the Sample Generate operation, in order to check it. (This operation should be done regardless of the results of the previous operation).

Solution: Use the same series of Parameter cards as in Problem 1, but insert a .TDP (Tape Dump) Order card with no ID code, before step 6.

Problem 3

After executing Problem 2, start a new sequence to assemble a new program and insert it in the ST. The second sequence is completely independent of the first, therefore requires a different ID code (BBBB).

Solution: Use the following sequence of Parameter cards:

No.	Operation Description	Parameter Cards	ID Code
1	Translate APS program	.APS Order card	AAA1
2	Introduce APS source program	(source program deck)	
3	Insert assembled program in ST	.API Order card	AAA1
4	Generate Sample File	.SGE Order card	AAA1
5	Introduce information to be recorded on tape	Detail cards	
	End of .SGE Order	End card	
6	Print contents of tape	.TDP Order card	ƀƀƀƀ
7	Load and start program	.PRL Order card	AAA1
8	Update File Table, if applicable	LBC Detail cards	
	End of .PRL Order	End card	
9	Translate APS program	.APS Order card	BBBB
10	Introduce APS source program	(source program deck)	
11	Insert assembled program in ST	.API Order card	BBBB

2. Pilot 0

This program is punched in a single card and is read into Core Store by normal manual initialization procedures. It searches the ST or MT for the first block of the Supervisor, called Pilot 1, thus starting ETOS.

Pilot 0 is essentially a fixed program, the only variable parameter being the name of the tape unit on which the ST or MT is mounted. It is stored permanently in the first 40 octets of SRA. The format of the Pilot 0 card is shown in the ETOS Operator's Manual, page III-9.

a. Restarting Pilot 0

During processing, Pilot 0 is restarted as follows:

1) After executing any program, a STOP macro-instruction causes a jump to the field in SRA (hexa.address 02A8) which resets the initial tape density and parity conditions. Then, if operating in STM (page III-16) Pilot 0 is restarted; if operating in MTM (page III-16), the Monitor is started.
2) If an error occurs during execution of a Type 1 or 3 program, the use of an ABORT macro-instruction causes a jump to the field in SRA which sets the ID switch to NO. The program then stops at address 02A6. By pressing START, Pilot 0 is restarted.

B. SUPERVISOR

This section contains a description of the Supervisor and its operation, as follows:

- Purpose and organization.
- Operation of Supervisor programs (Pilot 1, Pilot 2, Segment Load routine, BIOS).

- Condition checking during Supervisor operation. (See "GE-130 ETOS Operator's Manual" for details on operator actions).

The section also includes a description of Supervisor and System operation in STM and MTM.

1. Purpose and Organization

The Supervisor is the controlling part of ETOS, since it links the various phases of work, performs the physical Input/Output operations, and processes the Order programs. It is recorded in the ST and MT as the first Type 1 program, and consists of eight unlabelled blocks as shown below:

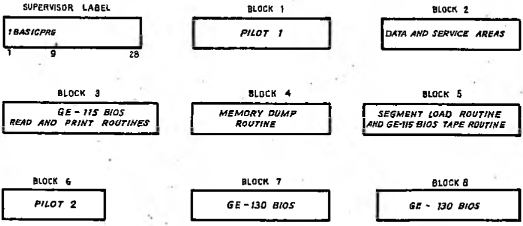


FIGURE 11 - FORMAT OF SUPERVISOR ON ST AND MT

Block 1: Pilot 1 is used to read and check the Order cards, and to start Pilot 2.

Block 2: The Data and Service areas consist of:

- All data describing the specific system configuration.
- ETOS service area.
- Card read area.
- Print area.

This block is stored in the SRA during ETOS operation.

Block 3: The GE-115 BIOS Read and Print Routines are variable, depending on the models of the Card Readers and Printers being used. This block is stored in the SRA whenever the use of 115 BIOS is specified in field C of the labels (see page II-3) of Type 1 or 3 programs.

Block 4: Is called into Core Store by Pilot 1 to execute a Memory Dump when the . MDP Order Card is read.

Block 5: The segment Load Routine searches for loads, and starts the first program segment. The 115 BIOS Tape Routine performs all physical I/O operations on Magnetic tapes.
This BIOS is stored in SRA whenever its use in specified in field C of the labels (see page II-3) of Type 1 or 3 programs.

Block 6: Pilot 2 starts a given program. It also introduces GE-120/130 BIOS into the SRA, when specified in field C of Type 1 or 3 program label.

Blocks 7 and 8: The 120/130 BIOS routines (where applicable) replace the 115 BIOS routines in the SRA, under control of Pilot 2 or the . PRL Order program.

2. Pilot 1

Pilot 1 is loaded in SRA and started by Pilot 0.

a. Reads Blocks 2 and 3 of the Supervisor into SRA.

b. Reads and processes Order cards as follows (see Flow Chart, fig. 12):

   1) Read a card

   2) Check to see that it is an Order card (valid card name in first four columns).

   3) If not, go back to 1).

   4) If an Order card, print five asterisks, today's date, and the contents of the Order card. (If a .CMT Order card, only print the associated comments).

   5) The ID Code is examined.

   6) If no ID Code, go to 15).

   7) If the ID Code is punched, check to see if it is the first Order card.

   8) If first Order card, move ID Code to ID Code Store Area in SRA, then go to 12).

   9) If not first Order card, compare ID Code to ID Code Store Area.

   10) If ID Codes equal, test the ID Switch in SRA (See page III-4), then go to 13).

   11) If ID Codes are different, move the new ID Code to ID Code Store Area.

   12) Set ID Switch to YES, then go to 15).

   13) If ID Switch set to NO, print 0196 * ORDER NOT EXECUTED and go back to 1).

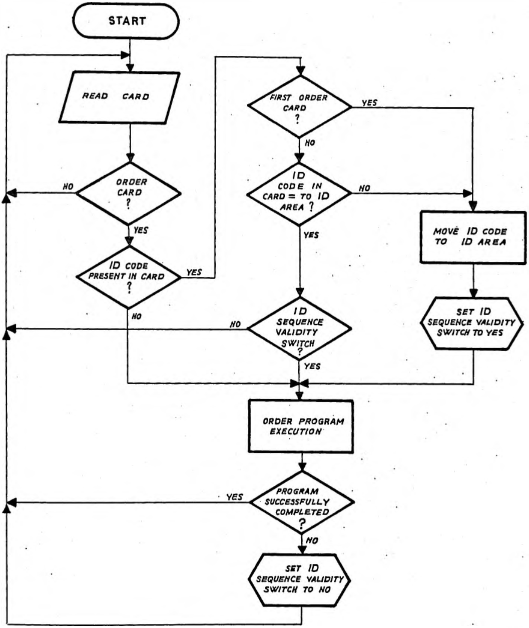

START
READ CARD
ORDER CARD ?
NO
YES
ID CODE PRESENT IN CARD ?
YES
NO
FIRST ORDER CARD ?
YES
NO
ID CODE IN CARD = TO ID AREA ?
NO
YES
MOVE ID CODE TO ID AREA
SET ID SEQUENCE VALIDITY SWITCH TO YES
ID SEQUENCE VALIDITY SWITCH ?
NO
YES
ORDER PROGRAM EXECUTION
PROGRAM SUCCESSFULLY COMPLETED ?
YES
NO
SET ID SEQUENCE VALIDITY SWITCH TO NO

FIGURE 12 - ORDER PROGRAM FLOWCHART

14) If ID Switch set to YES, go to 15).

15) If the Order is .CMT, go to 1); of the Prder is .HLT stop at address 01D6 (by pressing START, Pilot O can be restarted).

16) Read Block 4 of Supervisor (Memory dump Routine) in SRA.

17) If the Order is .MDP, this program is executed, then go back to 1).

18) If the Order is not .MDP

- Read and store Block 5 (Segment Load and 115 BIOS Tape routines) in SRA.
- Read and store Block 6 (Pilot 2) outside SRA.

19) Start Pilot 2.

NOTE: Pilot 1 is destroyed at the end of its execution.

3) Pilot 2

Pilot 2 operates in the following sequence:

a. Enter the ST or MT Select number in the Segment Load routine and location 0080 in SRA, then generate in SRA the PER instructions that reset the tape parity and density conditions.

b. Record blocks 7 and 8 (120/130 BIOS) outside SRA.

c. If the Order is .PRL, set the ID Switch to YES (a .PRL card with no or with different ID Code always interrupts a sequence of Order cards).

d. Search ST or MT for the Order program to be executed. If this search fails, print the message:

0297*ORDER UNKNOWN

and stop at address 02A6.
(By pressing START, Pilot 0 can be restarted).

e. If the Order program is found, print the program creation date (recorded in the program label).

f. If required, replace the 115 BIOS with the 120/130 BIOS in SRA.

g. If the Order is not . PRL, set URA to hexadecimal "0A".

h. Enter the program initial address in the Segment Load routine.

i. Start the Segment Load routine.

4. Segment Load Routine

The Segment Load routine searches the ST (or MT) for a program segment, loads it into Core Store and starts it.

This search is carried out by reading the segment numbers in program block labels, as follows:

a. Segment number found. The segment is loaded and started.

b. Segment number less than requested. Reading of block labels continues until either the required segment of the end-of-program TM is found. When reading a TM, the tape is rewound to the beginning-of-program TM, and the tape is searched forward again. If the required segment is not found, the tape stops at the end-of-program TM; the ID Switch is set to NO and processing stops at address 0200.

c. Segment number greater than requested. The routine rewinds the tape to the beginning-of-program TM and searches the tape forward.

If the required segment is not found, the tape stops at the end-of-program TM; the ID Switch is set to NO, and processing stops at address 0200.

d. End-of-program TM is read. The routine proceeds as in case c.

5. Execution of Order Programs

An Order program is executed on the basis of the parameters punched in the Order card and in the associated Detail cards, if any.

At the end of an Order program, control is transferred either:

a. To a user program, if the Order program was . PRL, or

b. To the Supervisor (which reads and processes the next Order).

During the execution of a user or Order program, the ID Switch may be set to NO as a result of some error condition, and control is transferred to the Supervisor.

6. Basic Input/Output System (BIOS)

a. General Concepts

BIOS is a set of routines for the physical processing of records on I/O media, such as the reading of a card or of a block on magnetic tape. BIOS is stored in SRA during system operation.

BIOS can be called by the APS source program, either directly (using BIOS macro-instructions) or indirectly (using EIOS macro-instructions). The BIOS macro-instructions are listed in paragraph c. below, while EIOS macro-instructions are described in the GE-120/GE-130 APS Manual.

b. Types of BIOS

The BIOS to be used under ETOS varies according to system configuration, as follows:

- 120/130 BIOS, with read/print routines
- 120/130 BIOS, with read/print/tape routines
- 115 BIOS, with read/print routines (see Note)
- 115 BIOS, with read/print/tape routines (see Note)

The type of BIOS required by each program is specified in the tenth character of the Type 1 or 3 Program Labels (fig. 3, page II-3).

In the case of Type 3 Programs on MT, the specified BIOS routines are incorporated into segment 00 of the program during MT generation (see .MTG Order). They are then transferred to SRA before program execution.

In the case of Type 1 Programs (either on ST or MT) or of Type 3 Programs (on ST only), the specified BIOS routines to be used in SRA are selected by the Supervisor, and loaded from ST (or MT), as required.

NOTE: The 115 BIOS may be used by the Type 1 programs and by those Type 3 Programs which were written for the GE-115 but not translated by the GE-120/130 Translator Programs.

c. BIOS Macro-Instructions

These are of two different types:

- Command, for the actual execution of I/O operations.

- Validation, for auxiliary operations, such as error checking, end-of-file detection, out-of-order checking.

The BIOS macro-instructions are listed below:

*READ = Read a card
*VCR = Validate Card Reading
*PRINT = Print a line and/or advance paper on Printer
*VPR = Validate Printout
*LBSS = Read and Print
*MTAPE = Operate Magnetic Tape Unit (read, write, erase or rewind)
*VMT = Validate Tape Operation (data transfer)
*TAPE = Operate Magnetic Tape Unit and Validate Operation

7. System Tape Mode (STM)

In STM all Order programs can be executed, since ST contains all of them. Type 3 programs can also be executed, but without programmed chaingin between programs (control is transferred to the Supervisor at the end of each program).

8. Master Tape Mode (MTM)

a. General Concepts

The MTM is the common user mode, as it provides for the normal execution of user programs.

In this mode, after the execution of each user program, control is transferred to its Monitor, which searches for, loads and starts the next program.

The Supervisor, .INT, and .PRL programs on the MT differ from the same programs on the ST in the following respects:

1) Supervisor

   a) In MTM, before reading an Order card, Pilot 1 checks the contents of the MT Control Area in SRA.

   b) If the MT Control Area is set to hexa. 00, another Order card is read and processed by Pilot 1 as described in steps 1 to 19 starting on page III-10.

   c) If the MT Control Area is set to 01, an Order card was previously read by the Monitor of the preceding program into SRA. The MT Control Area is reset to 00, and the Order card stored in SRA is processed by Pilot 1.

2) .INT

   In this program, all references to ST are changed to MT.

3) .PRL

   In MTM, before starting a user program, the .PRL program substitutes in the Re-entry Jump Area in SRA, the address of the Monitor of that User program. (In STM, the address of Pilot 0). Thus after executing a user program, control is transferred to the Monitor belonging to that program.

b. Monitor

The Monitor is created and inserted in the MT after each Type 3 program at MT Generation time (see .MTG Order). The Monitor format on tape is illustrated in fig. 9, page II-11.

After each Type 3 Program execution, the Monitor first checks the ID Switch. If it is set to NO, Pilot 0 is started, thus transferring control to the Supervisor. In any case, if the Routing Code in the first character of the Monitor is 4, the Master Tape Stop Order (.MTS Order) is executed.

If the Switch is set to YES, the Monitor prints the message

```
2299 * END OF PROGRAM X
```

then takes one of the following five courses, according to the Routing Code.

Routing Code	Monitor Operation
1	Print the message 2281*NEXT PROGRAM IS X ('X' was specified by the .MTG Order) Then search for, load and start the indicated program.
2	Print the message 2282 NEXT PROGRAM IS X WITH SWITCH 2 OFF/ Y WITH SWITCH 2 ON ('X' and 'Y' were specified by the .MTG Order) and stop at address 01D8. The Operator sets SWITCH 2 as desired then presses START. The Monitor stops again at address 01D8 to enable the Operator to reset SWITCH 2. After pressing START, the program specified in the message, is loaded and started.
3	Print the message 2283*NEXT PROGRAM SELECTED BY ORDER CARD Read an Order card. If it is not .PRL, start the Supervisor. If it is a .PRL card, the Monitor loads and starts the program specified by the .PRL card.

4	Execute the . MTS program, namely: rewind MT, print message 2284* MASTER TAPE STOP and stop at address 01D6. To restart Pilot 0, press START.
5	Print the message 2285 * NEXT PROGRAM X SELECTED BY ENDED PROGRAM Search for, load and start the program, the name of which was placed in the Read Area of SRA by the program just ended.

NOTES:

1) The Type 3 program seek time is reduced to a minimum by the use of the list of program names recorded in the Monitor (see Field F, figure 9, page II-11).
In fact, if the program to be started is in that list, the Monitor searches the tape backward. Otherwise, it executes a forward search.

2) A complete list of the Monitor messages is included in the . PRL Condition Checking Messages (Program Load from MT and Monitor), page III-50.

9. Supervisor Condition Checking

During execution of the Supervisor programs various condition checks are executed and corresponding messages may be printed out and/or the programs stopped at specific Core Store addresses.

a. Messages

Message Number	TITLE and NOTES	Stops at address
	During Pilot 1 Operation:	
0196	*ORDER NOT EXECUTED ID Switch set to NO. Pilot 1 looks for another Order card.	
	During Pilot 2 Operation:	
0201	*UNIT X: PERSISTENT ERROR Pilot 2 rewinds ST (or MT) and stops at address By pressing START, Pilot 0 is restarted.	02A6 (see 'b' following)
0202	*UNIT X: OUT OF ORDER Pilot 2 stops at address By pressing START, Pilot 0 is restarted.	02A6 (see 'b' following)
0297	*ORDER UNKNOWN Program specified by Order card not found on ST (or MT). Pilot 2 stops at address By pressing START, Pilot 0 is restarted.	02A6 (see 'b', following)

b. The Supervisor program may stop at one of the following HLT instructions:

HLT Address	Reason for Stop	Operator's Action
01CA	Persistent tape read error (during Pilot 1 operation)	Press START to make five more read attempts.
01D0	Unit out of Order (during Pilot 1 operation)	To restart Pilot 0, press OPERATE and START.
01D6	Pilot 1 is stopped by .HLT Order.	To read another Order card, press START.
02A6	Reason for Halt specified in message (see 'a' preceding). ID Switch set to NO.	To restart Pilot 0, press START.
0200	Segment not found by Segment Load Routine. ID Switch set to NO.	To restart Pilot 0, press START.
02CA	Persistent tape read error during execution of Segment Load Routine. ID Switch set to NO.	To restart Pilot 0, press START.

C. INDIVIDUAL ORDER PROGRAMS

The following pages give detailed descriptions of each Order. They are divided into five groups according to purpose, summarized as follows:

Group 1. INITIALIZE Orders

These are used to prepare the system for operation.

Group 2. TRANSLATE AND GO Orders

These call out the proper programs for:

- Translating symbolic language programs into machine language.
- Inserting the translated programs into the ST.
- Starting the translated programs.
- Punching object programs on cards.

Group 3. DEBUGGING Orders

These are used to:

- Set up a program for debugging.
- Generate sample files.
- Print the contents of a tape.
- Print the contents of Core Store.
- Duplicate tapes.
- List cards.

Group 4. PROGRAM TAPE GENERATE AND MAINTAIN Orders
These are used to generate, modify, and update the ST, LT, and MT.

Group 5. OTHER Orders
These call for miscellaneous service functions.

Each Order is normally described in the following sequence:

1. Purpose and Operation.

2. Parameter Cards, describing type, format and sequence of the Order, Detail, and End cards used.

3. Condition Checking Messages, listing all possible messages that may be printed out during the execution of the Order to notify the operator of the occurrence of some specific condition. When a condition stops the program, the address of the HLT instruction is displayed on the CPU console, and the proper message is printed out. For details of the corrective actions to be taken after each HLT instruction, see the ETOS Operator's Manual, listed in Appendix B of this book.

4. Example, showing a typical application of the Order.

NOTES:

- All abbreviations and special terms are defined in Appendix A, "Glossary of Terms", or in Appendix C, "SRA Chart".

- Tape units required for the individual Orders are specified in Appendix F.

1. INITIALIZATION ORDERS

a. .INT Initialize

1) Purpose and Operation

The program called by this Order performs the initial operations to start the system, as follows:

a) MT Control Area and ID Sequence Validity Switch are set to binary zero.

b) ID Code Store Area is set to 0.

c) The Day's date is loaded in SRA.

d) The USDA is loaded with the data punched in Order card cols. 39 - 70, if requested.

e) ST (or MT) and WT Unit Select numbers (1 - 8) are printed.

f) The numbers of the tape units containing WT's are loaded in Work Tape Unit Number area. (If WT3 is not used, location 0083 is set to hexadecimal FF).

The program also performs checks on the status of the tape units, as listed in Condition Checing Messages.

2) Parameter Cards

The program uses a single Order card, formatted as follows:

Columns	Contents
1 - 4	.INT
5 - 10	Today's date (b̸YYDDD)
11 - 16	Serial number of ST or MT if checking requested * in col. 11 if not requested.
17	Tape Unit number on which WT1 is mounted.
18 - 23	Serial number of WT1, if checking requested. * in col. 18 if rewind only. + in col. 18 if no operation.
24	Tape Unit number on which WT3 is mounted, if used. b̸ if not used.
32 - 37	Same as cols. 18 - 23, but for WT3, if used. b̸ if not used.
38	User Data Code: *if data to be loaded in USDA. b̸if not.
39 - 70	Contains data to be loaded in USDA, if requested.
71 - 80	Not used (ID Code not significant).

3) Condition Checking Messages

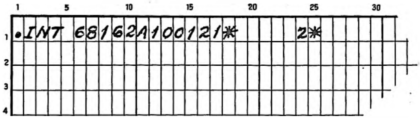
Message Number	TITLE and NOTES	Stops at address
1001	*UNIT X: PERSISTENT ERROR (in reading) if serial number check is requested	0000
1002	*UNIT X: OUT OF ORDER If after rewind command If after read command If during final rewind, program ends normally.	 01DA 02A6
1010	*UNIT X: SERIAL NUMBER Y WRONG/ SHOULD BE N	0000
1029	*INVALID DATE (cols. 5-10)	0000
1031	*USER DATA CODE WRONG (col. 38)	0000
1050	*UNIT NUMBER OF WT (1, 2, or 3) WRONG (cols. 17, 24, or 31)	0000
1051	*SYSTEM (or MASTER) TAPE AND WORK TAPE (1, 2, or 3) WITH SAME UNIT NUMBER	0000
1052	* WORK TAPE X AND WORK TAPE Y WITH SAME UNIT NUMBER	0000
1053	*UNITS OUT OF ORDER OR NOT SELECTED/ START	
1054	*UNITS OUT OF ORDER OR NOT SELECTED/ IMPOSSIBLE TO PROCEED	0000
1055	* TAPE ERROR/IMPOSSIBLE TO PROCEED Printed after Message 1001 or 1002	
1056	* SERIAL NUMBER WRONG/IMPOSSIBLE TO PROCEED Printed after message 1010	

1057	*TAPE ERROR AND SERIAL NUMBER WRONG/IMPOSSIBLE TO PROCEED Follows the messages 1001 or 1002, and 1010.	
1072	*SYSTEM (or MASTER) TAPE: UNIT X Designates unit on which tape is mounted.	
1073	*WORK TAPE 1: UNIT X WT1 is mounted on Unit X.	
1074	*WORK TAPE 2: UNIT X WT2 is mounted on Unit X.	
1075	*WORK TAPE 3: UNIT X WT3 not being used.	
1098	*IMPOSSIBLE TO PROCEED/LOAD PILOT 0 Printed after one of following messages: 1029, 1031, 1050, 1051, or 1052.	
1099	*END OF INITIALIZE	

4) Example

- Problem: To initialize the system.
  Date: Year, 1968; day 162
  ST Serial Number: A10012
  WT1 on Tape Unit 1
  WT2 on Tape Unit 2
  Rewind and check out-of-order conditions of WT's
  Data in USDA: None
  WT3 Used: No

- Solution: Punch Order Card as follows:



1
5
10
15
20
25
30
1
2
3
4

```
.INT 68162A100121*     2*
```

b. .TLW Tape Label Write

1) Purpose and Operation

This Order calls out the program for recording standard labels. It will record any number of tapes (except, of course, the MT or ST) as follows:

a) Read first Detail Card.

b) If correct, rewind tape, execute two erase instructions, and record labels and TM's.

c) If an error is found in a card, the program continues checking all parameters to disclose all possible errors in the card, but that tape is not labelled.

d) The program continues checking subsequent Detail Cards and writing the labels.

e) If errors were found in Detail Cards, the program ultimately stops at address 02A6.

f) Rewind tapes at end of program.

g) .TLW records the following information on each tape:

- Tape Header Label:
  VOL1 for 9-track tapes
  INT1 for 7-track tapes
- File Header Label: HDR1
- Two TM's

- Trailer Label:
  EOV1 if end of tape
  EOF1 if end of file
- Two TM's

2) Parameter Cards

The Order uses an Order card, one Detail card for each tape to be labelled, and an End card, formatted as follows:

ORDER CARD

Columns	Contents
1 - 4	.TLW
5 - 72	Not significant
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARD

1 - 4	b̸TLW
5	Punch 7 if 7-track Punch 9 if 9-track
6 - 7	W1 to label WT1 W2 to label WT2 W3 to label WT3 or Select number of unit containing tape to be labelled (b̸1 to b̸8).

8	Punch 0 for even parity. Punch P for odd parity.
9	Recording Density: K = 1600 bpi L = 800 bpi M = 556 bpi N = 200 bpi
10	Punch V if EOV1. Punch F if EOF1.
11 - 16	Tape Serial Number.
17 - 26	User's Name.
27 - 43	File Name.
44 - 49	Serial Number of first tape (volume) of file.
50 - 53	Sequence number of tape within file.
54 - 59	Labelling date (bYYDDD). If date not entered here, the day's date loaded by preceding .INT Order card is used. The day's date is always used as the purge date.
60 - 80	Not significant.

END CARD

1 - 4	ƀEND
5 - 80	Not significant.

.TLW

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at address
1101	*UNIT X: PERSISTENT ERROR (in recording)	02A6
1102	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after write command	02A6
	If during final rewind, program ends normally.	
1106	*INVALID DETAIL CARD Card name wrong. Tape not labelled.	02A6
1123	*TAPE UNIT CODE WRONG (cols. 6 - 7) Tape not labelled.	02A6
1126	*ILLEGAL PARITY CHECK CODE (col. 8) Tape not labelled.	02A6
1127	*ILLEGAL DENSITY CODE (col. 9) Tape not labelled.	02A6
1129	*INVALID DATE (cols. 54 - 59) Tape not labelled.	02A6

1130	*ILLEGAL TRAILER LABEL CODE (neither V nor F in Col. 10) Tape not labelled.	02A6
1135	*INVALID TRACKS NUMBER (neither 7 nor 9 in col. 5) Tape not labelled.	02A6
1199	*END OF TAPE LABEL WRITE	

4) Example

- Problem: To label WT1 and WT2
  WT1 = S/N 0A1251, Sequence number 0001.
  WT2 = S/N 0A1205, Sequence number 0001.
  Record in 9-track mode.
  Use 800 bpi, with odd parity.
  Record EOF1 labels.
  Use today's date (see .INT Order).

- Solution: Punch four cards as follows:

```
1   5    10   15   20  25   30 40   45   50
.TLW
 TLW9W1PLF0A1251 ISI    INVCE      0A12510001
 TLW9W2PLF0A1205 ISI    SVC        0A12050001
 END
```

2. "TRANSLATE and GO" Orders

a. .APS Assemble APS Program

1) Purpose and Operation

This Order calls out the APS Assembly program for translating from source to machine language. For details on the operation of the Order, see the GE-130 APS Manual; listed in Appendix B of this book.

2) Parameter Cards

The Order uses an Order Card only, formatted as follows:

Columns	Contents
1 - 4	.APS
5	ƀ if source program is on cards and in one of following languages: 130 APS 115 Basic APS 115 APS II 1 if source program is on cards and written in 115 Extended APS. 2 if source program is on WT1 and written in: 130 APS 115 APS II

Columns	Contents
6	ƀ if assembling program. 1 if assembling S/R.
7 - 14	Program or S/R name
15 - 49	Comments.
50 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

b. .COB Compile COBOL Program

1) Purpose and Operation

This Order calls out the COBOL Compiler for translating programs from source to machine language. For details on the operation of the Order, see the GE-120/130 COBOL Manual, listed in Appendix B of this book.

2) Parameter Cards

The Order uses a single Order card, formatted as follows:

Columns	Contents
1 - 4	.COB
5 - 8	Not significant
9	Punch T to indicate compile in a Tape System.
10	Punch L to request listing of object program. Otherwise ƀ.
11	Punch T is source program is on tape. ƀ if on cards.
12 - 71	Blank.
72	If T in col. 11, punch tape unit Select number containing source program.
73 - 80	Blank.

c. .FOR Compile FORTRAN Program

1) Purpose and Operation

This Order calls out the program which translates programs from source to machine language. For details on the operation of the Order, see the GE-130 FORTRAN Manual, listed in Appendix B of this book.

2) Parameter Cards

The Order uses a single Order Card, formatted as follows:

Columns	Contents
1 - 4	.FOR
5 - 8	Not significant.
9	Punch S if 7-track tape. Punch N if 9-track tape.
10	Punch T if source program is on tape. Punch C if on cards.
11	Punch L if object program listing is required. b̸ if not required.

Columns	Contents
12	Not significant.
13	3 (fixed character).
14	Indicates Core Store size for execution 3 = 12 K 4 = 16 K 5 = 24 K 6 = 32 K
15	Punch 1 if object program uses tapes. b̸ if tapes not used.
16	Blank.
17	Punch 1 if object program uses Card Punch. b̸ if Card Punch not used.
18 - 20	Indicates number of columns in Printer 104 = 104 columns 120 = 120 " b̸b̸b̸ = 136 "
21 - 25	Indicates initial address of object program in Core Store.
26 - 33	Program name.
34 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

d. .SPG Generate SORT/MERGE Program

1) Purpose and Operation

This Order calls out the SPG Generator which translates SPG source programs into APS language, which is then translated into machine language by an .APS Order. For details on the operation of the Order, see the GE-120/130 SPG Manual, listed in Appendix B of this book.

2) Parameter Cards

One Order card is required, formatted as follows:

Columns	Contents
1 - 4	.SPG
5 - 6	Blank
7 - 14	Name of program to be generated
15 - 49	Program description, to appear as the heading of program listing.
50	Blank, if the source program is on cards, 1, if the source program is on tape unit 1.
51	Blank, if there are no auxiliary sections or if they are on cards. 1, if they are on tape.
52	Tape recording density: 1 = 200 bpi 2 = 556 bpi 3 = 800 bpi 4 =1600 bpi
53	Blank if the parameters are on cards or on tape with odd parity, 1, if the parameters are on tape with even parity.
54-72	Blank
73-76	Optional ID Code
77-80	Blank

The .APS Parameter card (page III- 34) is used after the .SPG card with the following format:

Columns	Contents
1 - 4	.APS
5	Punch 2 to indicate source program is read from WT1.
6	ƀ to indicate that it is a program being assembled.
7 - 14	Name of program to be assembled.
15 - 49	Program description, to appear as the heading of program listing.
50 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

e. .API Assembled Program Insert

1) Purpose and Operation

This Order follows a translation Order and permits Type 1 or 3 program to be inserted into the ST. Also, whenever possible, it standardizes the length of program blocks to 4096 characters.

NOTES:
- The assembled Type 1 program is added o the end of existing Type 1 programs. In Type 3, the new program is inserted immediately after the "normal" programs (i. e., those not generated by the .DPG Order, which are stripped off the tape).
- Since the .APS Order loads the object program on WT2, the .API Order must be executed immediately afterwards, or after Orders which do not alter the contents of WT2.

The program is inserted into the tape as follows:

a) Rewind WT2.

b) On Type 1 programs:
- Position ST at beginning of S/R library.
- Write contents of ST on WT2 after assembled Type 1 program.
- Position ST after the Type 1 programs and copy from that point, standardizing block lengths of new program being inserted.

c) On Type 3 programs:
- Position ST after normal Type 3 programs, and before .DPG generated programs.
- Write new program on ST from that position, standardizing block lengths and deleting .DPG generated programs.

d) Rewind both tapes.

2) Parameter Cards

The Order uses an Order card only, formatted as follows:

Columns	Contents
1 - 4	.API
5	Program Type (1 or 3) of assembled program.
6 - 13	Name of program to be inserted
14 - 72	Not significant
73 - 76	Optional ID code
77 - 80	Blank.

3) Condition Checking Messages

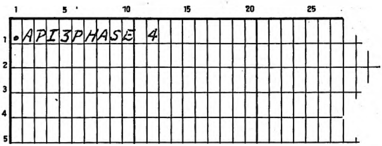
Message Number	TITLE and NOTES	Stops at Address
2001	*UNIT X: PERSISTENT ERROR (read or write)	02A6
2002	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program ends normally.	
	*PROGRAM X NOT FOUND (on WT2) (cols. 6 - 13)	02A6
2013	*MULTIPLE PROGRAM NAME TX (already on ST) New program is inserted, old program to be removed with .STS Order.	02A6
2021	*ILLEGAL PROGRAM TYPE (col. 5)	02A6
2038	*INVALID LABEL (on WT2)	02A6
2071	*PROGRAM LABEL...... (Label recorded on WT2 is printed)	
2099	*END OF ASSEMBLED PROGRAM INSERT	

.API

4) Example

- <u>Problem:</u> Insert Type 3 program, PHASEƀ4, in the ST. The program was assembled by the .APS Order.

- Solution: The .API Order card is formatted as follows:



```
.API3PHASE 4
```

1
5
10
15
20
25
1
2
3
4
5

f. .PRL Program Load

1) Purpose and Operation

This Order calls a program which searches for, loads into Core Store, and starts a Type 3 program located in the ST or MT. In addition, if the program is on the ST, it may be executed with Path Trace Routine (printing all jump instructions which have occurred). It is also possible to load and start a program generated for debugging (see .DPG Order) which is in the ST.

The program consists of:
    Program Loader
    Path Trace Routine

The sequence of operations is as follows:

a) Load Trace Routine into Core Store, if requested.

b) Search for the desired program and load the associated File Table into Core Store.

c) Read the LBC cards, if any, and update the File Table.

d) Load 'OA' (hexadecimal) in all available Core Store locations, if requested.

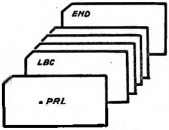
e) Load user data into USDA, if requested.

f) Call the Segment Load routine which loads the first program segment and starts processing.

g) Before program start-up a HLT occurs at address 01D2, if requested. In this case press START on CPU console to begin processing.

2) Parameter Cards

The Order and End cards are always used. The LBC cards are used only if the File Table is to be updated. The cards are formatted as follows:



ORDER CARD

Columns	Contents
1 - 4	.PRL
5 - 12	Name of program to be loaded.
13	Punch C if program to be loaded is generated by .DPG ƀ if otherwise
14	Punch P if Trace Routine requested. ƀ if otherwise
15	Punch H if HLT to be executed before starting program. ƀ if otherwise
16	b if 'OA' to be loaded in available store. *if otherwise.
17	*if data to be transferred to USDA. ƀ if otherwise
18 - 49	Data to be transferred to USDA.
50 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank

LBC CARD

The LBC card format is described in Appendix D since it is used in other Orders.

. PRL

END CARD

Columns	Contents
1 - 4	ƀEND
5 - 80	Not significant

3) Condition Checking Messages

FOR ST

Messages Number	TITLE and NOTES	Stops at Address
2101	*UNIT X: PERSISTENT ERROR (in reading)	02A6
2102	*UNIT X: OUT OF ORDER If after read command If during final rewind, the program ends normally.	02A6
2106	*INVALID DETAIL CARD	02A6
2108	*PROGRAM TX NOT FOUND	02A6
2120	*INVALID PROGRAM NAME (cols. 5 - 12 of LBC Card or col. 13 or . PRL Card). See NOTE at end of Table.	02A6
2124	* MAIN UNIT CODE WRONG (LBC Card, col. 43). See NOTE at end of Table.	02A6
2125	* ALTERNATE UNIT CODE WRONG (LBC Card, col. 44). See NOTE at end a of Table.	02A6

2127	*ILLEGAL DENSITY CODE (LBC Card, col. 51). See NOTE at end of Table.	02A6
2128	*FILE IDENTIFICATION CODE X NOT FOUND (In File Table) (LBC Card, cols. 13-18). See NOTE at end of Table.	02A6
2129	*INVALID DATE (on LBC Card). See NOTE at end of Table.	02A6
2131	*USER DATA CODE WRONG (PRL Card, col. 17)	02A6
2132	*MEMORY LOADING CODE WRONG (PRL Card, col. 16)	02A6
2133	*HALT CODE WRONG (PRL Card, col. 15)	02A6
2134	*INVALID FILE CHECK OPTION CODE (LBC Card, cols. 45-50). See NOTE at end of Table.	02A6
2136	*FILE TABLE OVERFLOW Indicates one of the following conditions: 1)File table starting address less than 12 K and final address greater than 5376 decimal. 2)File table starting address greater than or equal to 12 K and final address beyond usable Core Store capacity.	02A6
2137	*FILE TABLE NOT FOUND Program to be loaded has no File Table. Followed by message 2106.	02A6
2150	*TRACE ROUTINE CODE WRONG (PRL Card, col. 14)	02A6
2151	*EXECUTION WITH TRACE ROUTINE IMPOSSIBLE Program starts anyway	
2199	*END OF PROGRAM LOAD	

NOTE: When error is in LBC card, the program finishes checking that card, before stopping at address 02A6.

2201	*UNIT X: PERSISTENT ERROR Press START to reload Pilot 0.	02A6
2202	*UNIT X: OUT OF ORDER Press START to reload Pilot 0. If message 2284 follows, ID sequence Vaildity Switch is not changed.	02A6 01D6
2206	*INVALID DETAIL CARD (LBC Card, cols. 1 - 4) MT is rewound. Press START to reload Pilot 0 See Note at end of Table.	02A6
2208	*PROGRAM X NOT FOUND (.PRL Card, cols. 5 - 12) MT is rewpund. Press START to reload Pilot 0	02A6
2220	*INVALID PROGRAM NAME (LBC Card, cols. 5 - 12) (same as for message 2206)	02A6
2224	* MAIN UNIT CODE WRONG (LBC Card, col. 43) (same as for message 2206)	02A6
2225	*ALTERNATE UNIT CODE WRONG (LBC Card, col. 44) (same as for message 2206)	02A6
2227	*ILLEGAL DENSITY CODE (LBC Card, col. 51) (same as for message 2206)	02A6

2228	*FILE IDENTIFICATION CODE X NOT FOUND (in File Table) (LBC Card, cols. 13 - 18) (same as for message 2206)	02A6
2229	*INVALID DATE (LBC Card) (same as for message 2206)	02A6
2231	*USER DATA CODE WRONG (. PRL Card, col. 17) MT is rewound. Pilot 0 reloaded by pressing START	02A6
2232	*MEMORY LOADING CODE WRONG (. PRL Card; col. 16) MT is rewound. Pilot 0 reloaded by pressing START.	02A6
2233	*HALT CODE WRONG (. PRL Card, col. 15) MT is rewound. Pilot 0 reloaded by pressing START.	02A6
2234	*INVALID FILE CHECK OPTION CODE (LBC Card, col. 45 - 50) (same as for message 2206)	02A6
2236	*FILE TABLE OVERFLOW Indicates one of the following conditions: 1) File table starting address less than 12 K and final address greater than 5376 decimal. 2) File Table starting address greater than or equal to 12 K and final address beyond usable Core Store capacity.	02A6
2237	*FILE TABLE NOT FOUND Program to be loaded has no File Table. (Followed by message 2206)	02A6
2280	*YYDDD/START OF PROGRAM X Indicates beginning of program	

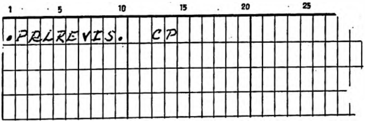
. PRL

2281	*NEXT PROGRAM IS X	
2282	*NEXT PROGRAM IS X WITH SWITCH 2 OFF/Y SWITCH 2 ON	01D8
2283	*NEXT PROGRAM SELECTED BY ORDER CARD	
2264	*MASTER TAPE STOP Program is ended, MT is rewound.	01D6
2285	*NEXT PROGRAM X SELECTED BY ENDED PROGRAM	
2299	*END OF PROGRAM X	

NOTE: When error is in LBC card, program finishes checking that card before stopping at address 02A6.

4) Example

- Problem: Program, REVIS., generated by .DPG is to be loaded from ST. Path Trace Routine printout required. Available Core Store area set to 'OA'. No data to be loaded, and no LBC Cards to process.
- Solution: Punch card as follows:



```
1   5    10   15   20   25
.PRLREVIS.  CP
```

g. .APP Assembled Program Punch

1) Purpose and Operation

This Order is used to punch any Type 1 or 3 program recorded on the ST, or any program assembled on WT2 by the .APS Order (page III-34).
The sequence of operations is as follows:

a) Select the proper Punch routine (for the type of Punch used in the system).

b) Rewind tape and search for the desired program.

c) Punch program in hexadecimal and print.

The program also punches a PLC Card (with an I or an R in col. 36) to begin the deck; a SIN Card to begin each segment; and an END Card to finish the program, this when obtained can be used for any subsequent updating of the ST (see .STU order, page III-99).

.APP

2) Parameter Cards

The Order uses an Order card only, formatted as follows:

Columns	Contents
1 - 4	.APP
5	Type (1 or 3) or program to be punched
6 - 13	Name of program to be punched.
14	ƀ if program to be punched is on ST. * if program is on WT2.
15 - 18	Program code to be punched in Cols. 1 - 4 of output cards.
19	ƀ if col. 36 of PLC Card is to be punched I. R if col. 36 of PLC Card is to be punched R.
20	* if the SIN or END card following a program segment is to be punched with the quantity (in decimal) of cards containing the segment. ƀ if otherwise.
21 - 72	Not significant
73 - 76	Optional ID Code.
77 - 80	Blank

3) Condition Checking Messages

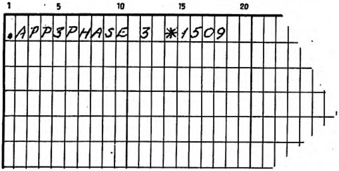
Message Number	TITLE and NOTES	Stops at Address
2301	*UNIT X: PERSISTENT ERROR (read)	02A6
2302	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read command	02A6
	If during final rewind, program ends normally.	
2307	*PUNCH ERROR ON** CARD The incorrect card is identified in the card listing by two asterisks. Operator must repunch card correctly.	02A6
2308	*PROGRAM TX NOT FOUND (on ST or WT2)	02A6
2315	*PUNCH NOT CONNECTED	02A6
2321	*ILLEGAL PROGRAM TYPE (col. 5, .APP Card)	02A6
2323	*TAPE UNIT CODE WRONG (col. 14, .APP Card)	02A6
2338	*INVALID LABEL (WT2 program label)	02A6
2346	*ILLEGAL PLC TYPE CODE (col. 19, .APP Card)	02A6
2350	*ILLEGAL CARD COUNT CODE (col. 29 of .APP Card)	02A6

2371	*PROGRAM LABEL:. . . . . (if program is punched from WT2, label is printed).	
2399	*END OF ASSEMBLED PROGRAM PUNCH	

4) Example

- Problem: A program assembled by the .APS Order, is to be punched from WT2.
  Name: PHASEƀ3
  Program Code: 1509
  Program Type: 3
  PLC Card, col. 36: Punch I

- Solution: Punch the following Order Card:



```
1   5    10   15   20
.APP3PHASE 3 *1509
```

3. DEBUGGING ORDERS

a. .TDC Tape Duplicate

1) Purpose and Operation

The Order is used to duplicate (either in part or all) any tape containing standard or non-standard data files, or to duplicate a program tape (ST, LT, or MT). The output tape may retain none, part, or all of the blocks previously recorded on it.

The program operates in the following sequence:

a) Load input and output tapes to proper intial positions.

b) Duplicate.

NOTE: Any CHKP blocks on the input tape are disregarded by the .TDC program (see figure page III-154).

2) Parameter Cards

The Order uses an Order Card only, punched in the following format:

. TDC

Columns	Contents
1 - 4	. TDC
5 - 6	Type of file on input tape. ƀP if tape is ST, LT, or MT DS if tape contains standard data file. DN if tape contains non-standard data file.
7 - 8	W1 if input tape is WT1. W2 if input tape is WT2. W3 if input tape is WT3. or the Select number of the unit containing the input tape (ƀ 1 to ƀ 8).
9	Parity of input tape O = Even P = Odd
10	Recording density on input tape K = 1600 bpi L = 800 bpi M = 556 bpi N = 200 bpi
11 - 16	Serial number of input tape, if it is to be checked. Otherwise, blank.
17 - 18	Number of tape unit containing output tape (see cols. 7 - 8 above).
19	Parity of output tape (see col. 9).
20	Recording density on output tape (see col. 10).
21 - 26	Serial number of output tape, if it is to be checked. Otherwise, blank.

27	Punch C if file on input tape must follow a file already on output tape (permitted only if 'DS' in cols. 5 - 6). Otherwise, blank.
28 - 39	File name, only if it must be changed on output tape (permitted only if 'DS' in cols. 5 - 6 and col. 27 is blank). Otherwise, blank.
40 - 45	File serial number,only if it must be changed on output tape (same restrictions as for File Name). Otherwise, blank.
46 - 48	Output tape retention period. Blank if purge date is today's date. (Dates in HDR1 label are updated only if 'DS' in cols. 5 - 6 and col. 27 is blank).
49 - 53	Quantity of initial blocks on input tape not to be duplicated. Blank if all blocks to be duplicated. (Significant only for non-standard files; i. e., 'DN' in cols. 5 - 6).
54 - 58	Quantity of intial blocks previously recorded on output tape to be retained. Blank if none. (In this case, two Erase operations are executed). (Significant only for non-standard files; i. e., 'DN' in cols. 5 - 6).
59	Punch N if blocks are not numbered. (Significant only if 'C' in col. 27). Otherwise, blank.

.TDC

60	V, to close output tape with EOV1. Any other character, to close output tape with EOF1.
61 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
5101	*UNIT X: PERSISTENT ERROR (read or write)	02A6
5102	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program ends normally.	
5110	*UNIT X: SERIAL NUMBER Y WRONG/SHOULD BE N	02A6
5116	*UNIT X: END OF REEL (output tape) 1) Duplication completed. 2) Duplication incomplete. Output tape closed by: . TM and Program Tape Trailer Label, if bP in cols. 5 - 6. . TM, if 'DN' in cols 5 - 6. . TM and EOF1 label (with incomplete block count) if 'DS' in cols. 5 - 6.	02A6

5117	*UNIT X: NON-STANDARD FILE (if 'bP' or 'DS' in cols. 5 - 6)	02A6
5150	*FILE TYPE CODE WRONG (cols. 5 - 6)	02A6
5151	*INPUT TAPE UNIT CODE WRONG (cols. 7 - 8)	02A6
5152	*ILLEGAL INPUT PARITY CHECK CODE (col. 9)	02A6
5153	*ILLEGAL INPUT DENSITY CODE (col. 10)	02A6
5154	*OUTPUT TAPE UNIT CODE WRONG (cols. 17 - 18)	02A6
5135	*ILLEGAL OUTPUT PARITY CHECK CODE (col. 19)	02A6
5156	*ILLEGAL OUTPUT DENSITY CODE (col. 20)	02A6
5157	*INVALID FILE APPENDING CODE (col. 27)	02A6
5158	*INVALID RETENTION PERIOD (cols. 46 - 48)	02A6
5159	*INVALID NUMBER OF BLOCKS ON INPUT TAPE (cols. 49 - 53)	02A6

5160	*INVALID NUMBER OF BLOCKS ON OUTPUT TAPE (cols. 54 - 58)	02A6
5199	*END OF TAPE DUPLICATE	

4. Example

- Problem: Tape duplication
  Input Tape Specifications
  Contents: Standard file
  Tape Unit Select Number: 2
  Parity: Odd
  Recording Density: 800 bpi
  Tape Serial Number: 000015 (tape to be checked)

  Output Tape Specifications
  Tape Unit Select Number: 3
  Parity and Density: Unchanged
  Tape Serial Number: 000024 (tape to be checked)
  File Serial Number and Name: Unchanged
  Purge Date: Today
  Input Tape Blocks to be Recorded: All
  Output Tape Blocks to be Retained: None

- Solution: Punch Order card as follows:

```
1   5    10   15   20   25   30   35
.TDCDS 2PL000015 3PL000024
```

b. .DPG Debugging Program Generate

1) Purpose and Operation

The Order is used for setting up controlled execution (debugging) of a Type 3 program on the ST, by inserting corrections in machine language, and by providing a Memory Trace Routine for printing out Core Store areas at specified trace points.

The quantity of corrections and the use of the Memory Trace Routine is dependent upon Core Store availability. The Trace Routine must occupy a store area external to the Type 3 program area. Its length is 466 octets.

A trace point table is loaded after the Trace Routine, and its length varies with the number of trace points requested. (The table consists of N records of 15 octets each). In specifying trace points, note that: a) Each trace point requires four octets to receive a jump instruction to the Trace Routine. Therefore the trace points must be specified at addresses containing 4- or 6-octet instructions. b) The jump instruction destroys the previous contents of Register 7 (locations 254-255).
c) It is not permissible to insert trace points in macro-instructions or in the corrections introduced by CRT cards (see below).

The modified program is written at the end of the Type 3 program library in the ST, and is identified by the character C in the 10th position of its label. The .DPG program operates as follows:

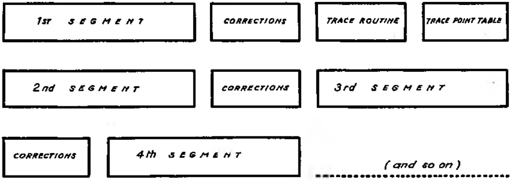
a) Search ST for program specified by Order card.

b) Write program on WT1, inserting any corrections (CRT cards) at the end of each segment.

c) Transfer corrected program from WT1 to ST.

d) If Trace Routine is requested (DTR Card), it is inserted at end of first segment of Type 3 program, and after any corrections that have been made. The jump instructions (TRC Cards) are inserted in the corrected program. The trace point table is recorded after the Trace Routine. (This table contains Memory Trace specifications (TRC Cards) and directs the operation for each trace point in Type 3 programs).

At the end of the .DPG Order the Type 3 program will have the following Debugging Format on ST:



2) Parameter Cards

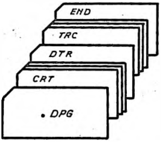
.DPG uses an Order card, three types of Detail cards, and an END card. The Detail cards are used in the following sequence:

CRT (Correction Card): when used, they are arranged in ascending segment numbers.

DTR (Define Trace Card): When used, only one is required for the program.

TRC (Trace Point Card): When used, one card is needed for each trace point and they are inserted in card deck in ascending segment numbers, and by ascending addresses within each segment.

The parameter cards are formatted as follows:



ORDER CARD

Columns	Contents
1 - 4	.DPG
5 - 12	Name of program to be prepared for debugging
13	* if all programs generated by previous .DPG orders are to be deleted from ST. Otherwise, blank.
14 - 72	Not significant
73 - 76	Optional ID Code.
77 - 80	Blank.

.DPG

CRT CARD

Columns	Contents
1 - 4	ƀCRT
5 - 6	Length of correction (punched in hexadecimal).
7 - 10	Address in Core Store where correction is made (punched in hexadecimal).
11 - 72	Correction in machine language (punched in hexadecimal).
73 - 78	Not significant.
79 - 80	Number (in decimal) of segment affected by correction.

DTR CARD

1 - 4	ƀDTR
5 - 8	Start address of Trace Routine in Core Store.
9 - 80	Not significant.

TRC CARD

1 - 4	ƀTRC
5 - 6	Decimal number of segment where trace point is required.

Columns	Contents
7 - 10	Hexadecimal trace point address in Core Store where jump instruction to Trace Routine is to be inserted.
11 - 14	Initial address (in hexadecimal) of store area to be printed.
15 - 18	Length (in hexadecimal) of store area to be printed.
19 - 80	Not significant.

END CARD

1 - 4	b̸END
5 - 80	Not significant.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
3001	* UNIT X: PERSISTENT ERROR (read, write or verify)	02A6
3002	* UNIT X: OUT OF ORDER If after rewind command If after read or write command If during final rewind, program ends normally.	 01DA 02A6

Message Number	TITLE and NOTES	Stops at Address
3006	* INVALID DETAIL CARD	02A6
3008	* PROGRAM TX NOT FOUND (on ST)	02A6
3013	* MULTIPLE PROGRAM NAME TX ST contains another debugging format program with the same name. The Order program proceeds to its end before stopping at address	02A6
3041	* DATA CARD WRONG (refers to last one read and printed)	02A6
3042	* SEGMENT NUMBER X WRONG (cols. 79 - 80 of CRT, cols. 5 - 6 of TRC) Segment number in descending order, or not in decimal.	02A6
3043	* SEGMENT NUMBER X NOT FOUND	02A6
3044	DATA LENGTH WRONG (cols. 5 - 6 of CRT, cols. 15 - 18 of TRC)	02A6
3050	* INVALID ADDRESS (on CRT, DTR or TRC, or an odd address is specified on DTR)	02A6
3051	* GENERATED PROGRAM OVERFLOW (in Core Store)	02A6
3052	* TRC TABLE OVERFLOW (Trace point table too long for available store)	02A6

3053	* TRC POINT X WRONG (Length of instruction at trace point address is two octets only)	02A6
3054	* PROGRAM DELETION CODE WRONG (col. 13 of .DPG)	02A6
3099	* END OF DEBUGGING PROGRAM GENERATE	

4) Example

- <u>Problem:</u> To correct in machine language the second segment of program, PHASE𝑏3, and to delete all programs generated by previous .DPG orders.
  Length of correction: 13 octets.
  Correction Address: 0C09
  Correction to be made: (in hexadecimal)
  (as indicated in columns 11 to 36 in CRT Card, below)

- <u>Solution:</u> Punch Order, CRT, and END Cards as follows:

```
1   5   10   15   20   25   30   35                                      75   80
.DPGPHASE 3 *
 CRT0D0C09F40201078407FED20607F60F58                                          02
 END
```

Rev. Nov. 1969

c. .SGE Samples Generate

1) Purpose and Operation

This Order permits the recording of a sample file on a WT, from data contained in Detail cards. Standard or non-standard files with numbered or un-numbered blocks can be written.

If a standard file on a labelled tape is desired, the program automatically closes the file. If there are variable length records, the lengths must be indicated by the user in the first two characters of each record. The maximum block lengths depend on Core Store capacity, as follows:

Core Store	Block Length
12K	7K
16K	11K
24K	19K
32K	27K

The program operates in the following sequence:

a) Rewind the WT.

b) Position the WT following the blocks to be retained. If there are none, execute two erase operations.

c) Read, print and record the Detail cards on the WT.

d) Close the file as specified by the End card, and rewind both the ST and the WT.

2) Parameter Cards

.SGE uses an Order card, four types of Detail cards, and an End card. The Detail cards are used as follows:

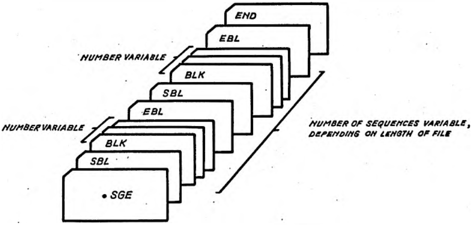
SBL (Start Block) One for each block.

BLK (Block) These cards contain the data to be recorded in the block. The number of BLK cards will vary

EBL (End Block) Signals the end of the BLK cards. One for each block.

TPM (Tape Mark) Used to record a TM where required.

The parameter cards are formatted as follows:



.SGE

ORDER CARD

Columns	Contents
1 - 4	.SGE
5 - 6	W1 if output tape is WT1. W2 if output tape is WT2. W3 if output tape is WT3. Or Select number of unit containing the output tape (ƀ1 to ƀ8).
7	Parity Check Code O = Even P = Odd
8	Recording Density Code K = 1600 bpi L = 800 bpi M = 556 bpi N = 200 bpi
9	Indicates character type on BLK Detail cards. X = Characters punched in hexadecimal (applicable only on 9-track tape). Alphameric characters can be inter-mixed by punching a period (.) before each alpha character. Therefore, each two characters (either alpha or hexa) are recorded as one character on tape. ƀ = Alphanumeric characters only. (Each single character is recorded as a character on tape).
10 - 14	Quantity of blocks to be retained on WT. To obtain a standard file on a labelled tape, it is necessary to retain at least three blocks.

Columns	Contents
15 - 72	Not significant
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARDS

	SBL (Start Block)
1 - 4	ƀSBL
5 - 9	Sequence number of block (in decimal) to be recorded. ƀ if block is not numbered.
10 - 80	Not significant.

	BLK (Block)
1 - 4	ƀBLK
5 - 6	Number of characters (in decimal) to be recorded on WT.
7 - 80	The characters to be recorded.

	EBL (End Block)
1 - 4	ƀEBL
5 - 80	Not significant.

.SGE

Columns	Contents
	TPM (Tape Mark)
1 - 4	ƀTPM
5 - 80	Not significant

END CARD

1 - 4	ƀEND
5	Specifies type of file close. V = End of Colume (EOV1) F = End of File (EOF1). b = File already closed by Detail card. The End card causes tape to be rewound. NOTE: File closing can be done by the .SGE program only if it is a standard tape, and the Order card has requested retention of at least three blocks.
6 - 80	Not significant

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at address
3101	*UNIT X: PERSISTENT ERROR (read or write)	02A6
3102	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program ends normally.	
3106	*INVALID DETAIL CARD	02A6
3112	*UNIT X: HDR1 LABEL NOT FOUND (on WT) Program continues normally to END card.	
	If no 'b̸' in col. 5 of End card, program stops at address	02A6
3116	*UNIT X: END OF REEL Tape is closed with TM, EOF1 label, and rewound.	02A6
3123	*TAPE UNIT CODE WRONG (cols. 5 - 6, Order Card)	02A6
3126	*ILLEGAL PARITY CHECK CODE (col. 7, Order card)	02A6
3127	*ILLEGAL DENSITY CODE (col. 8, Order card)	02A6
3130	*ILLEGAL TRAILER LABEL CODE (col. 5, End card)	02A6

3144	*DATA LENGTH WRONG Number of characters to be recorded (cols. 5 - 6 of BLK) not in decimal, or greater than 74, if ƀ in col. 9 of . SGE " " 37, if X " " "	02A6
3145	*BLOCK LENGTH WRONG Length of block is greater than allowed by available Core Store.	02A6
3147	*BLOCK NUMBER WRONG Number expressed in cols. 5 - 9, SBL card. Quantity expressed in cols. 10 - 14 . SGE card.	02A6
3150	*DATA RECORDING MODE WRONG (col. 9 . SGE card)	02A6
3199	*END OF SAMPLES GENERATE	

4) Example

. Problem: To record a sample file of one block length on WT1. Standard file with numbered blocks. Close file with EOF1 label.
Parity: Odd
Recording density: 800 bpi
Data punched in hexadecimal
Blocks to be retained: 3
(SBL) Block sequence number : 00001

- **Solution**: Punch cards as follows:

```
      1   5    10   15   20   25   30
  1   .SGEW1PLX00003
  2    SBL00001
  N  { BLK  }(Cards containing Block)
     { BLK  }
N+3    EBL
N+4    ENDF
```

d. .TDP Tape Dump

1) Purpose and Operation

This Order permits the printout of a specified portion of a tape, either in hexadecimal or in alphameric characters. The maximum block lengths depend on Core Store capacity, as follows:

Core Store	Block Length
12K	7K
16K	11K
24K	19K
32K	27K

The Order operates in the following sequence:

a) Rewind tape to the position preceding first block to be printed.

b) Print the following heading:
3277 *TAPE DUMP OF UNIT X FROM BLOCK Y/BLOCKS Z
(See Condition Checking Messages).

c) Print a line at the head of each block, showing block sequence number and quantity of characters it contains.

d) The program stops after printing the number of blocks specified in the Order card; or after printing the Trailer Label if the quantity of blocks is not specified.

e) Rewind tape.

2) Parameter Cards

The Order used an Order card only, formatted as follows:

Columns	Contents
1 - 4	.TDP
5 - 6	W1 to print WT1. W2 to print WT2. W3 to print WT3. ST to print ST (or MT) Or Select number of unit containing tape to be printed (b̸1 to b̸8).
7	Parity Check Code O = Even P = Odd
8	Recording Density K = 1600 bpi L = 800 bpi M = 556 bpi N = 200 bpi
9	Specifies type of printout X = Print hexadecimal b̸ = Print alphanumeric
10 - 14	Number of first block to be printed. If not indicated, printing starts at first block on the tape. On a standard tape, this would be at the label, VOL1 or INT1.

15 - 19	Quantity of blocks to be printed. If not indicated, all blocks will be printed and a Trailer Label will be printed at the end.
20 - 72	Not significant
73 - 76	Optional ID Code.
77 - 80	Blank.

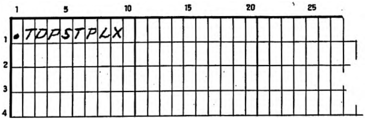
3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
3201	*UNIT X: PERSISTENT ERROR (read) Tape is rewound and Pilot 0 is restarted.	
3202	*UNIT X: OUT OF ORDER If after rewind command If after read command If during final rewind, program ends normally.	 01DA 02A6
3223	*TAPE UNIT CODE WRONG (cols. 5 - 6)	02A6
3226	*ILLEGAL PARITY CHECK CODE	02A6
3227	*ILLEGAL DENSITY CODE (col. 8)	02A6

3247	*BLOCK NUMBER WRONG Number expressed in cols. 10 - 14 and/ or quantity expressed in cols. 15 - 19.	02A6
3250	*PRINTOUT MODE WRONG (col. 9)	02A6
3277	*TAPE DUMP OF UNIT X FROM BLOCK Y/BLOCKS Z X = Select number of unit on which tape to be printed is mounted. Y = Serial number of first block to be printed. Z = Quantity of blocks to be printed.	
3299	*END OF TAPE DUMP	

4) Example

- . **Problem:** To print the contents of the ST.
  Parity Check Code: Off
  Recording Density: 800 bpi
  Type of printout: Hexadecimal
- . **Solution:** Punch the Order Card as follows:



```
.TDPSTPLX
```

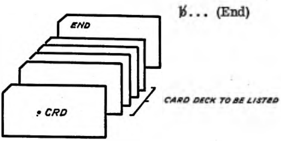
e. .CRD Card Dump

1) Purpose and Operation

This program is used to execute a listing of all parameter cards in a given Order, to check for correctness of the deck. The program reads and prints the cards simultaneously, providing an 80/80 listing of each card.

2) Parameter Cards

The program uses an Order card and an End card, formatted as follows:



b̸... (End)

ORDER CARD

Columns	Contents
1 - 4	.CRD
5 - 72	Not significant
73 - 76	Optional ID Code.
77 - 80	Blank

END CARD

Columns	Contents
1 - 4	ƀ... (To distinguish this End card from End card of deck being listed).
5 - 80	Not significant.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
3302	*UNIT X: OUT OF ORDER (during rewind) Program ends normally.	
3399	*END OF CARD DUMP	

f. .MDP Memory Dump

1) Purpose and Operation

The program permits the printout of a specified area in Core Store, both in the SRA, and in the area available for user programs. The SRA remains unchanged except for the printer reserved area (hexadecimal address 0200 to 02A1) and the area occupied by the .MDP program (hexadecimal 02BC to 0474).

The program is contained in block 4 of the Supervisor and is started by Pilot 1. Each line is printed in hexadecimal and is followed by a line printed in graphic characters.

2) Parameter Cards

The Order requires an Order card only, formatted as follows:

Columns	Contents
1 - 4	.MDP
5 - 8	Initial address (in hexadecimal) of Core Store area to be printed ɓ if printout is to start at initial address of area available to user.
9 - 12	Final address of Core Store area to be printed. ɓ if Core Store is to be printed up to its final address.

Columns	Contents
13 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank

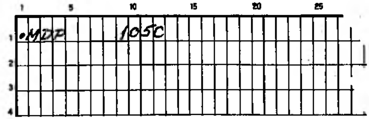
3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
9095	* MEMORY DUMP OF PROGRAM X Name of last program loaded into Core Store. This message appears at the head of the printout.	
9099	* END OF MEMORY DUMP	

4) Example

- **Problem:** To printout Core Store contents from the initial address of the area available to the user, through end address 105C (hexadecimal).

- **Solution:** Punch the Order card as follows:



.MDP 105C

g. .PCC Punched Card Check

1) Purpose and Operation

This Order is used to verify the program cards punched by the .APP Order, against the original program on ST or WT2.

The program operates as follows:

a) Search ST or WT2 for required program.

b) Contents of program cards are compared with the program on tape.

c) If discrepancies are detected, the incorrect card(s), identified by two asterisks, are printed out, together with the corresponding information on tape.

2) Parameter Cards

The Order uses an Order card only, which is followed by the program card deck punched by the .APP Program. See page III-53, paragraph under c).

The Order card format is shown in the following table.

Columns	Contents
1 - 4	.PCC
5	Type (1 or 3) of program to be checked.
6 - 13	Name of program to be checked.
14	ƀ if original program is on ST. * if original program is on WT2.
15 - 18	Program code, punched in cols. 1 - 4 of program cards.
19	ƀ if col. 36 of PLC card contains a "I". R if col. 36 of PLC card contains an "R".
20	* if the SIN or END card following a program segment specifies the quantity (in decimal) of the cards containing the segment. ƀ if otherwise.
21 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

. PCC

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at address
5801	*UNIT X: PERSISTENT ERROR (read)	02A6
5802	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read command	02A6
	If during final rewind, program ends normally.	
5808	*PROGRAM TX NOT FOUND (on ST or WT2)	02A6
5821	*ILLEGAL PROGRAM TYPE (col. 5, . PCC card)	02A6
5823	*TAPE UNIT CODE WRONG (col. 14, . PCC card)	02A6
5846	*ILLEGAL PLC TYPE CODE (col. 19, . PCC card)	02A6
5850	*ILLEGAL CARD COUNT CODE (col. 20, . PCC card)	02A6
5851	*ERROR ON CARD Incorrect card(s), identified by two asterisks, are printed out, together with the corresponding information on tape. Incorrect characters are indicated by arrows. Program continues to end, but stops at address.	02A6
5899	*END OF PUNCHED CARD CHECK	

4) Example

- Problem: Verify the program card deck punched by the .APP Order (see page III-56).

- Solution: Punch the following Order card:

```
1   5    10   15   20   25
.PCC3PHASE 3 *1509
    Followed by card deck
      punched by .APP Order
```

h. . PSC Punched Subroutine Check

1) Purpose and Operation

This Order is used to verify the program cards punched by the . SBP Order, against the original information on ST or WT2.

The program operation is similar to that of the . SBP program, but results in a comparison rather than in punching/printing. It also performs the same checking operations as the . SBP program.

If discrepancies are detected between the contents of cards and the original information on tape, the incorrect card(s), identified by two asterisks, are printed out, together with the corresponding information on tape.

2) Parameter Cards

The program uses an Order card only, formatted as follows:

Columns	Contents
1 - 4	. PSC
5 - 10	Name of S/R to be checked. * in col. 5 if entire S/R Library is to be checked.
11	* if S/R was punched from WT2. ƀ if S/R was punched from ST. This column is significant when checking one S/R only.
12 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at address
5501	* UNIT X: PERSISTENT ERROR (read or write)	02A6
5502	* UNIT X: OUT OF ORDER	
	If after a rewind command	01DA
	If after a read or write command	02A6
	If during final rewind, program ends normally.	

. PSC

5507	*PUNCH ERROR ON **CARD Incorrect card(s), identified by two asterisks, are printed out together with the corresponding information on tape. Program continues to end, but stops at address.	02A6
5509	* SUBROUTINE YX NOT FOUND (on ST or WT2)	02A6
5523	* TAPE UNIT CODE WRONG (col. 11, . PSC card)	02A6
5538	* INVALID LABEL Label generated by APS Assembler not found on WT2.	02A6
5539	* INVALID SUBROUTINE NAME (cols. 5-10 of Order card) S/R name does not commence with Y.	02A6
5550	* SUBROUTINE ADDRESS WRONG The initial address of the S/R is not 0000.	02A6
5551	*SOURCE STATEMENT WRONG A programming restriction has been disregarded (see "Restrictions", page III-118). The wrong instructions are printed out.	02A6
5599	*END OF PUNCHED SUBROUTINE CHECK	

4) Example

- Problem: To verify the S/R punched on cards by the . SBP Order (see page III-56. 5).
- Solution: Punch the Order card as follows:

```
.PSCYPNSTA*                                                             BBBB
  Seguita dal pacco schede
 perforato dall'Ordine .SBP
```

i. . PSI Punched Subroutine Insert

1) Purpose and Operation

This Order permits one or more S/R's to be inserted in the Library on ST, using the cards punched by the . SBP Order.

It operates as follows:

a) Read S/R Index from ST into WT1.

b) Read S/R(s) from cards into WT2.

c) Update S/R Index on WT1.

d) When End card is read, write contents of ST and WT2 on WT1 inserting S/R(s) at the end of S/R Library.

e) Copy WT1 to ST.

2) Parameter Cards

The program uses an Order card, one or more Detail cards (SSB) and an End card. Each SSB card is followed by the program cards containing the S/R to be inserted. These cards are described under . SBP, page III-56. 2. The Parameter cards format is shown in the following tables.

.PSI

ORDER CARD

Columns	Contents
1 - 4	.PSI
5 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARD

	SSB (Start Subroutine)
1 - 4	ȻSSB
5 - 6	Length, in hexadecimal, of information punched in cols. 17 - 52.
7 - 12	Name of S/R to be inserted.
13 - 16	Length of S/R, in hexadecimal.
17 - 52	Names of S/R(s) called out by the S/R to be inserted (max. six S/R's).
53 - 80	Not significant.

END CARD

1 - 4	ȻEND
5 - 80	Not significant.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at address
5601	*UNIT X: PERSISTENT ERROR (read or write)	02A6
5602	*UNIT X: OUT OF ORDER	
	If after a rewind command	01DA
	If after a read or write command	02A6
	If during final rewind, program ends normally.	
5606	*INVALID DETAIL CARD Detail card(s) missing or out of sequence.	02A6
5609	*SUBROUTINE YX NOT FOUND The S/R YX called out by the S/R specified by the previous SSB card, is not present in the library on ST. This S/R is not inserted but the program continues to end, then stops at the address.	02A6
5611	* CARD COUNT WRONG Cards containing S/R do not have consecutive ascending sequence numbers (cols. 13 - 16 of card described on page III-56. 2). The S/R is not inserted but the program continues to end, then stops at the address.	02A6

. PSI

5614	*MULTIPLE SUBROUTINE NAME YX S/R specified for insertion is already in Library. The program processes the other cards, then stops at the address	02A6
5629	*INVALID DATE (cols. 7-12 of S/R card, page III-56.2) The S/R is not inserted, but the program processes the other cards, then stops at the address.	02A6
5639	*INVALID SUBROUTINE NAME SSB and associated S/R cards have different S/R name, or not commencing with Y. The S/R is not inserted, but the program continues to end, then stops at the address.	02A6
5641	*DATA CARD WRONG Information in cols. 19-80 of S/R card (page III-56.2) not in valid form. S/R not inserted, but program continues to end, then stops at address.	02A6
5644	*DATA LENGTH WRONG (cols. 17-18 of S/R card, page III-56.2) S/R not inserted but program continues to end, then stops at address.	02A6
5699	*END OF PUNCHED SUBROUTINE INSERT	

4) Example

- **Problem**: To insert in Library on ST the S/R punched on cards by the . SBP Order, and verified by the . PSC Order, (see page III-56.5)

- **Solution**: Punch the following Order card:

```
.PSI
```

Seguita dal pacco schede
perforato da .SBP:
scheda SSB, schede S/R, scheda END.

j. .TCK System Tape Checking

1) Purpose and Operation

This program is used to check the new System Tape generated on WT2 from the current ST and WT1 by means of the .TPU program.

The program is in five segments, operating as follows:

- Segment 1 (Read first Detail card and call appropriate segment)

Same operation as described under .TPU (page III-150.8), except that two block read areas are provided, each occupying 4096 Core Store locations.

- Segment 2 (check ST duplication)
  Compare previous ST with WT2.

- Segment 3 (check Operating System replacement)
  Compare Operating Systems on WT1 and WT2.
  Compare Type 3 programs on ST and WT2.

- Segment 4 (Read and store Detail cards)
  Same operation as described under .TPU, page III-150.9.

- Segment 5 (Check updated System Tape)

a) Rewind ST and check labels (if requested).

b) Check WT2, comparing its contents with previous ST and WT1, and signal which blocks are invalid, and why. This validity check has obviously some limitations which cannot be stated here. In any case the program provides the number of any error block (header labels and TM's are also counted), both for WT2 and the tape used for comparison, thus allowing, by means of the .TDP (Tape Dump) program, to inspect tapes starting from the blocks signalled.

c) Print out the unprocessed Detail cards (see message 7741). They contain some errors which cannot be detected during program execution.

d) Rewind tapes.

2) Parameter Cards

The .TCK program requires an Order card, up to four types of Detail cards (described below) and an End card.

PIN (Program Insert) is used to check the insertion of a new program from WT1.

PRR (Program Replace) is used to check the replacement of a program on ST with an updated version on WT1.

PRD (Program Delete) is used to check the deletion of a program or a series of consecutive programs on ST.

ØSR (Operating System Replace) is used to check the replacement of the Operating System on ST (except for the Supervisor) with an updated version on WT1.

The Parameter cards are formatted as follows:

ORDER CARD

Columns	Contents
1 - 4	.TCK
5 - 10	Serial number of ST, if checking requested. Otherwise blank.
11 - 16	Serial number of WT1, if checking requested. Otherwise blank.
17 - 22	Serial number of WT2, if checking requested. Otherwise blank.
23 - 72	Not significant
73 - 76	Blank
77 - 80	Blank.

DETAIL CARDS

The PIN, PRR, PRD and ØSR Detail cards have the same format as used for the .TPU Order (see pages starting from III-150.12).

END CARD

Same as for .TPU Order.

3) Condition Checking Messages.

Message Number	TITLE and NOTES	Stops at address
7701	* UNIT X: PERSISTENT ERROR (read)	02A6
7702	* UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program ends normally.	
7705	* PROCEEDING IMPOSSIBLE	02A6
7706	* INVALID DETAIL CARD Name incorrect (cols. 1 - 4). Program checks the other Detail cards. At the end, message 7705 is printed out and program stops at address	02A6
7708	* PROGRAM TX NOT FOUND (on ST or WT)	02A6
7710	* UNIT X: SERIAL NUMBER X WRONG/ SHOULD BE Y	02A6
7713	* MULTIPLE PROGRAM NAME TX Program to be inserted is already present on ST.	02A6
7719	* DETAIL CARD OVERFLOW/X LAST CARD Program continues and processes only the accepted Detail cards, then stops at address	02A6

7721	* ILLEGAL PROGRAM TYPE (cols. 5 and/or 14 of Detail cards) Program checks the other Detail cards. At the end, message 7705 is printed out and program stops at address	02A6
7741	* DATA CARD WRONG Followed by printout of incorrect card	02A6
7760	* OUTPUT BLOCK X WRONG X is block sequence number One of the following additional specifications may appear: - PROGRAM TX NOT FOUND (on ST or WT1) - BLOCK LENGTH WRONG - BLOCK DATA WRONG - MISSING BLOCKS Tape Mark encountered on one tape only.	02A6
7779	* SYSTEM TAPE INDEX Followed by listing of programs on new System Tape. Either of the following specifications may appear: - DUPLICATE if duplication from ST to WT2 was requested. - DUPLICATE: PROGRAMS TYPE 1 AND 2 FROM WT1 AND PROGRAMS TYPE 3 FROM ST. Replacement of operating system was requested.	
7799	* END OF SYSTEM TAPE CHECKING	

4) Example

- Problem: To check System Tape generated on WT2 in the example given for the .TPU Order (see page III-150.18).

- Solution: Punch cards as follows:

```
.TCK
 PRD1A
 PRD1C        1L
 PRD3P        3R
 PRR1.APS
 PRR1L
 PRR3R
 PIN1E        1BASICPRG
 PIN1F        1L
 PIN1I        1F
 PININ        1I
 PIN1K
 PIN1H
 PIN30        2BASICSUB
 PRD1B
 PIN1B        1.APS
 END
```

Comparisons are made between WT2 and ST or WT1 depending on the origin of WT2 programs.

4. PROGRAM TAPE GENERATE AND MAINTAIN ORDERS

a. .STG System Tape Generate

1) Purpose and Operation

This Order is used to generate the ST from the Sample ST, for a specific system configuration. The three Orders to create a new ST are :

.INT – Initialize
.TLW – Tape Label Write (on Work Tapes)
.STG – System Tape Generate (on WT1)

The .STG program adapts the Supervisor to the system configuration by:

- Generating new blocks to be inserted in the Supervisor.
- Modifying the BIOS routines contained in the .MTG program (on the sample ST).

The new Supervisor blocks and modified BIOS routines are recorded on WT2. Then the .STG generates the ST on WT1 by duplicating the Sample ST and WT2.

2) Parameter Cards

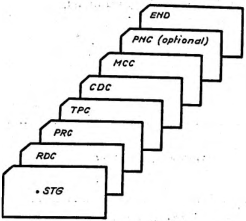
This program requires an Order card, at least five Detail cards to define the system configuration, and an End card. The required Detail cards are :

RDC – Model of Card Reader
PRC – Model of Printer
TPC – Type of Tape Controller
CDC – Type of Card Code
MCC – Core Store Capacity

If the system includes a Card Punch, an additional Detail card is required:

PNC - Model of Card Punch

The cards are formatted as follows:



ORDER CARD

Columns	Contents
1 - 4	.STG
5	Recording Density Code K = 1600 bpi L = 800 bpi M = 560 bpi
6 - 11	Serial number of WT1, if check is required. Otherwise blank.
12 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARDS

	RDC - (Reader Compile)
1 - 4	￠RDC
5 - 6	Model of Reader ￠1 = CRZ100, CRZ111, or CRZ120 ￠2 = CRP100

Columns	Contents
7 - 9	Peripheral Unit Name (in decimal). Blank if CRZ100, CRZ111 or CRZ120.
10 - 80	Not significant.

	PRC - (Printer Compile)
1 - 4	ƀPRC
5 - 6	Model of Printer ƀ1 = PRT100 or PRT110 ƀ2 = PRT120 or PRT130 ƀ3 = (to be assigned) ƀ4 = (to be assigned) ƀ5 = (to be assigned)
7 - 9	Peripheral Name (in decimal). Blank if PRT100 or PRT110.
10 - 80	Not significant.

	TPC - (Tape Compile)
1 - 4	ƀTPC
5 - 6	Type of Controller ƀ1 = (to be assigned) ƀ2 = 7/9 track Controller ƀ3 = (to be assigned) ƀ4 = Phase Modulation Controller ƀ5 = (to be assigned) ƀ6 = (to be assigned)

.STG

Columns	Contents
7	Number of tape units – from 3 to 8 (in decimal)
8 – 9	(To be assigned)
10 – 12	(To be assigned)
13 – 80	Not significant.

	CDC – (Code Compile)
1 – 4	b̸CDC
5	H (for Hollerith code). B (for Bull code).
6 – 80	Not significant

	MCC – (Memory Capacity Compile)
1 – 4	b̸MCC
5 – 6	Core Store capacity Punch 12, 16, 24 or 32.
7 – 80	Not Significant

Columns	Contents
	PNC - (Punch Compile)
1 - 4	bPNC
5 - 6	Model of Card Punch b1 = CRP100 b2 = CPZ101 b3 = CPZ103
7 - 9	Peripheral Unit Name (in decimal).
10 - 80	Not significant.

END CARD

1 - 4	bEND
5 - 80	Not significant

.STG

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
4001	*UNIT X: PERSISTENT ERROR (read or write)	02A6
4202	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program end normally.	
4010	*UNIT X: SERIAL NUMBER Y WRONG/SHOULD BE N Order card wrong. S/N in cols. 6 - 11 does not match S/N in WT1.	02A6
4020	*INVALID DETAIL CARD	02A6
4027	*ILLEGAL DENSITY CODE (col. 5, Order card)	02A6
4040	*INVALID PERIPHERAL NAME X (cols. 7 - 9 of RDC, PRC, PNC) Name not in decimal or greater than 255.	02A6
4050	*DETAIL CARD MISSING One or more of 5 cards missing.	02A6
4051	*INVALID CORE STORE CAPACITY CODE (cols. 5 - 6 of MCC card)	02A6
4052	*INVALID PERIPHERAL CODE (cols. 5 - 6 of RDC, PRC, TPC, PNC cards)	02A6
4053	*CONFIGURATION NOT ACCEPTED Insertion in Supervisor has been requested for a routine not recognized by the .STG program.	02A6
4099	*END OF SYSTEM TAPE GENERATE	

4) Example

- <u>Problem</u>: To generate a ST on WT1 (previously labelled) for following configuration:
  Density Code: 800 bpi
  Reader: CRZ100
  Printer: PRT100
  Tape Controller: 7/9-track, 3 tape units
  Core Store capacity: 16K

- <u>Solution</u>: Punch cards as follows:

```
1   5    10   15   20   25
.STGL
 RDC 1
 PRC 1
 TPC 23
 CDCH
 MCC16
 END
```

b. .SPD System Tape Print and Duplicate

1) Purpose and Operation

This program permits the duplication of the ST (or MT) while printing out the labels of all porgrams in the tape. Either printing or duplication is optional, but in duplication, programs in Debugging Format Mode (see .DPG order) are eliminated.

The program operates in the following sequence:

a) The tape(s) are rewound and positioned after the TM's following the two Header labels.

b) The ST or MT is duplicated and/or labels are printed.

c) If there is an *in column 8, a TM and trailer label are recorded after the S/R Library. If column 8 is blank the Type 3 programs are also printed and/or duplicated.

2) Parameter Cards

The program uses an Order card only, formatted as follows:

Columns	Contents
1 - 4	.SPD
5	Execution Type Code b̸ = Duplicate and Print D = Duplicate only P = Print only

6 - 7	W1 = to duplicate on WT1 W2 = to duplicate on WT2 W3 = to duplicate on WT3 Or, Tape Unit Select Number on which duplication will be done (b̸1 to b̸8).
8	Program Type Code b̸ If Type 3 programs also to be duplicated/printed * Otherwise.
9 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
4101	*UNIT X: PERSISTENT ERROR (read or write)	02A6
4102	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program ends normally.	
4118	*EXECUTION TYPE CODE WRONG (col. 5)	02A6

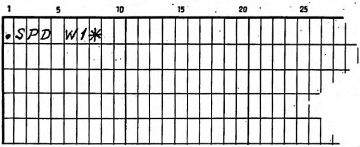
.SPD

4123	*TAPE UNIT CODE WRONG (cols. 6 - 7) Either non-existent unit number, or same unit number as ST or MT is mounted on.	02A6
4151	*USER PROGRAM CODE WRONG (col. 8)	02A6
4199	*END OF SYSTEM TAPE PRINT AND DUPLICATE	

4) Example

- Problem: Duplicate on WT1 the ST, excluding all Type 3 programs.
  Print labels.

- Solution: Punch the Order card as follows:



```
.SPD W1*
```

c. .STC System Tape Close

1) Purpose and Operation

This program is used to eliminate from the ST the Type 3 programs which are no longer needed. As specified in column 5 of the Order card, the ST will be positioned after the S/R Library, or after the normal Type 3 program specified by the user (columns 6 - 13). A TM and the trailer label are then recorded and the tape is rewound.

2) Parameter Cards

The program uses an Order card only, formatted as follows:

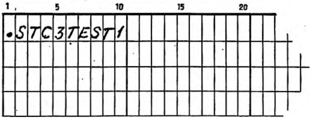
Columns	Contents
1 - 4	.STC
5	2 = Close ST after S/R Library 3 = Close ST after a Type 3 program
6 - 13	Name of Type 3 program. Used only when '3' in col. 5.
14 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
4201	*UNIT X: PERSISTENT ERROR (read or write)	02A6
4202	*UNIT X: OUT OF ORDER If after read or write command If during final rewind, program ends normally.	02A6
4208	*PROGRAM TX NOT FOUND (X in cols. 6 - 13)	02A6
4221	*ILLEGAL PROGRAM TYPE (col. 5)	02A6
4299	*END OF SYSTEM TAPE CLOSE	

4) Example

- Problem: To close the ST after a Type 3 program.
  Name of program: TEST1
- Solution: Punch the Order card as follows



```
.STC3TEST1
```

d. .STU System Tape Update

1) Purpose and Operation

This program is used to update the ST by:

- Writing new, or updating existing program labels (see figure 3, page II - 3).
- Inserting new segments, while deleting pre-existing segments from the insertion point up to the end of the program.
- Replacing existing segments.

The program always eliminates any Debugging Format Mode programs (see .DPG Order, page III-63) from the ST.

2) Parameter Cards

The program uses an Order card, up to three types of Detail cards, and an End card. The Detail cards are:

PLC - for writing a new, or updating an existing program label.
SIN - for inserting a new segment.
SRP - for replacing an existing segment.

Each SIN or SRP card is followed by its associated program cards. All SIN and SRP cards must be arranged in ascending segment numbers.

Each .STU Order updates one program only, and the associated Detail cards must refer to that program. The Order may be used in any of the following configurations:

a) Insert new segment(s) only
.STU + SIN(s) + END

b) Replace existing segment(s) only
.STU + SRP(s) + END

c) Insert new program label
   .STU + PLC (with 'I' in col. 36) + END

d) Insert new program label and segment(s) in program
   .STU + PLC (with 'I' in col. 36) + SIN(s) + END

e) Replace existing program label
   .STU + PLC (with 'R' in col. 36) + END

f) Replace existing program label and insert new segment(s) in program
   .STU + PLC (with 'R' in col. 36) + SIN(s) + END

g) Replace existing program label and existing segment(s) in program
   .STU + PLC (with 'R' in col. 36) + SRP(s) + END

The Parameter cards are formatted as follows:

ORDER CARD

Columns	Contents
1 - 4	.STU
5 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARDS

Columns	Contents
	PLC (Program Label Create)
1 - 4	ƀPLC
5 - 13	Type (1 or 3) and name of program whose label is to be created (new or substitute label).
14	Type of BIOS required by program.
15 - 23	If 'I' in col. 36 , type and name of program after which new label is to be inserted. If cols. 15 - 23 are blank, the new label will be inserted after all Type 1 or 3 programs (as indicated in col. 5). If 'R' in col. 36, type and name of program whose label is to be replaced. If cols. 15 - 23 are blank, the label of the program indicated in cols. 5 - 13 will be updated without changing the program name.
24 - 27	Initial address of File Table (in hexadecimal). If no table, indicate address 0000.
28 - 31	Initial address of program (in hexadecimal).
32 - 35	First Core Store location available after end of program (in hexadecimal).
36	I if new label is to be inserted. (In this case, the PLC card is normally followed by SIN card(s) in order to generate a new program on ST. If no SIN card(s); a TM is written after the label). R if label is to be replaced.

37 - 42	Date (ƀYYDDD) to be inserted in program label. Blank if today's date to be inserted.
43 - 80	Not significant.

	SIN (Segment Insert)
1 - 4	ƀSIN
5 - 13	Type and name of a program in which new segment is to be inserted. If a new program, this field must be the same as that indicated in preceding PLC card, cols. 5 - 13.
14 - 15	Number of segment after which insertion is to be made. 00 if it is the first program segment. (If more than one segment is to be inserted in the same program, they are inserted in sequence since the segment number of the first SIN card only is recognized by the . STU program).
16 - 19	Not significant.
20 - 21	Number to be assigned to segment being inserted (in decimal).
22 - 32	Not significant.
33 - 36	Indicates quantity of program cards (in decimal) used by the preceding SIN card, if any. (To check that no program cards are missing as tape is loaded). If check not required, these columns are blank. If first SIN card, these columns are not significant. If one SIN card only, or if last one of series, the quantity of associated program cards is punched in cols. 33 - 36 of END card.

37 - 42	Date (b̸YYDDD). Date on first SIN card only is significant. Date on PLC card, if any, overrides this date. Blank if today's date is used.
43 - 80	Not significant.

	SRP (Segment Replace)
1 - 4	b̸SRP
5 - 13	Type (1 or 3) and name of program in which the substitution is to be made.
14 - 15	Number (in decimal) of the segment to be replaced.
16 - 32	Not significant.
33 - 36	Number (in decimal) of program cards with preceding SRP (if any). This number is checked during tape loading. If check is not required, these columns are not punched. To check the last SRP, this quantity is punched in cols. 33 - 36 of the END card.
37 - 42	Date (bYYDDD). Date on first SRP only is significant. Date on PLC card, if any, overrides this date. Blank if today's date is used.
43 - 80	Not significant.

.STU

END CARD

Columns	Contents
1 - 4	ƀEND
5 - 32	Not significant.
33 - 36	Number of program cards with last SIN or SRP card, to be checked during tape load. Blank if check not required.
37 - 42	Date (ƀYYDDD). Blank if today's date is used. Date on PLC card, if any, overrides this date.
43 - 80	Not significant.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
4301	*UNIT X: PERSISTENT ERROR (read or write)	02A6
4302	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program ends normally.	
4306	*INVALID DETAIL CARD Detail card with invalid name or out of sequence.	02A6
4308	*PROGRAM TX NOT FOUND (on ST) (cols. 15 - 23, PLC card) or (cols. 5 - 13, SIN or SRP cards)	02A6

4311	*CARD COUNT WRONG (cols. 33 - 36, SIN, SRP, or END cards)	02A6
4313	*MULTIPLE PROGRAM NAME TX Program found on ST with same name as program to be inserted (PLC with 'I' in col. 36). Program is inserted anyway.	02A6
4320	*INVALID PROGRAM NAME col. 14 of PLC card wrong, or Detail card with different program name (cols. 5 - 13).	02A6
4321	*ILLEGAL PROGRAM TYPE	02A6
4329	*INVALID DATE	02A6
4341	*DATA CARD WRONG Incorrect program code, or invalid data (on program card).	02A6
4342	*SEGMENT NUMBER X WRONG Not in ascending sequence, or invalid.	02A6
4343	*SEGMENT NUMBER X NOT FOUND (on ST)	02A6
4346	*ILLEGAL PLC TYPE CODE Neither 'I' nor 'R' in col. 36.	02A6
4399	*END OF SYSTEMTAPE UPDATE	

.STU

4) Example

- Problem: To insert a new program into the ST as the last Type 3 program.
  New program name: PHASEb3

	Decimal	Hexa
File Table address:	2400	0960
Initial program address:	2432	0980
First address after end of program:	5696	1640

  Phaseb3 uses data files on tape, and consists of a single segment contained in 268 cards.

- Solution: Punch cards as follows:

```
     1   5    10   15   20   25   30   35
1    .STU
2     PLC3PHASE 3 5         0960098016401
3     SIN3PHASE 3 00    01
      }268 Program Cards
272   END           0268
```

e. .STS System Tape Strip

1) Purpose and Operation

This program is used to remove Type 1 or 3 programs from the ST, operating as follows:

a) Position WT1 after the Header labels.

b) Read and print Detail cards which specify programs to be eliminated from ST.

c) Search ST for first program to be removed.

d) Transfer contents of ST to WT1, excluding the programs to be removed.

e) Re-position the ST to the point ahead of the first program removed, and transcribe contents of WT1 to ST.

f) Print names of programs to be removed which were not found on ST.

g) Rewind both tapes.

2) Parameter Cards

The program requires an Order card, one Detail card (in any sequence) for each program to be removed, and an End card formatted as follows:

ORDER CARD

Columns	Contents
1 - 4	.STS
5 - 72	Not significant
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARD

1 - 4	␢STS
5 - 13	Type (1 or 3) and name of program to be removed.
14	C if program to be removed is in Debugging Format Mode (see .DPG Order) ␢ otherwise
15 - 80	Not significant.

END CARD

1 - 4	␢END
5 - 80	Not significant.

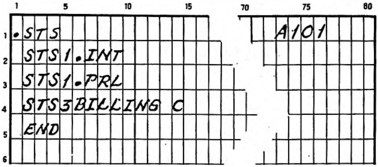
3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
4401	*UNIT X: PERSISTENT ERROR (read or write)	02A6
4402	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program ends normally.	
4406	*INVALID DETAIL CARD (cols. 1 - 4)	02A6
4408	*PROGRAM TX NOT FOUND (on ST) Card not processed but program continues to end, and stops at address	02A6
4420	*INVALID PROGRAM NAME (Neither 'C' nor 'b' in col. 14 of Detail card for Type 3 programs). Card not processed but program continues to end, and stops at address	02A6
4421	*ILLEGAL PROGRAM TYPE (col. 5, Detail card) Card not processed but program continues to end, and stops at address	02A6
4499	*END OF SYSTEM TAPE STRIP	

4) Example

- Problem: To remove three programs from ST.
  Program types and names:
  1 .INT
  1 .PRL
  3 BILLING
  BILLING is in Debugging Format Mode (.DPG)
  .STS Order belongs to ID sequence A101.

- Solution: Punche cards as follows:



```
.STS                                                                    A101
 STS1.INT
 STS1.PRL
 STS3BILLING C
 END
```

f. . SLS Subroutine Library Strip

1) Purpose and Operation

This program is used to remove from one to all S/R's from the ST, and to print a new listing(Index) of the remaining S/R's. It also updates the S/R Index in the APS Assembler. Up to 640 S/R's can be removed.

It operates in the following sequence:

a) Read and print the Detail cards which specify the S/R's to be removed.

b) Position WT1 after the Header labels, and the ST ahead of the block in the APS Assembler where the S/R Index is recorded.

c) Update the S/R Index.

d) Rewind and duplicate the ST on WT1, removing S/R's as required.

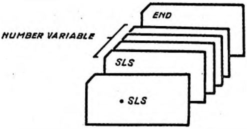
e) Print new S/R Index to check that all S/R's called for have been found and removed.

f) Transcribe WT1 to ST and rewind both tapes.

2) Parameter Cards

The program uses an Order card, one Detail card for each S /R to be removed (inserted in the deck in any sequence), and an End card. If the entire S/R Library is to be removed, not Detail cards are required.

The cards are formatted as follows:



ORDER CARD

Columns	Contents
1 - 4	.SLS
5	*If entire S/R Library to be removed. b̸ Otherwise
6 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARD

1 - 4	b̸SLS
5 - 10	Name of S/R to be removed.
11 - 80	Not significant.

END CARD

1 - 4	ƀEND
5 - 80	Not significant.

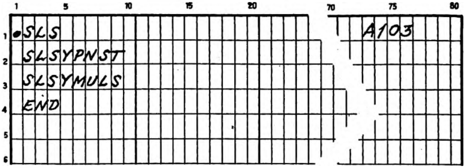
3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
4501	* UNIT X: PERSISTENT ERROR (read or write)	02A6
4502	* UNIT X: OUT OF ORDER If after rewind command If after read or write command If during final rewind, program ends normally.	 01DA 02A6
4506	* INVALID DETAIL CARD Error in cols. 1 - 4 of Detail card, or End card follows Order card when there is a ƀ in col. 5 of Order card.	02A6
4509	* SUBROUTINE YX NOT FOUND (on ST) Program continues to end but stops at address	02A6

Message Number	TITLE and NOTES	Stops at Address
4518	* EXECUTION TYPE CODE WRONG (col. 5 of Order card)	02A6
4539	* INVALID SUBROUTINE NAME (cols. 5 - 10 of Detail card) S/R name does not begin with Y. Card not processed, program continues to end but stops at address	02A6
4578	* SUBROUTINE LIBRARY INDEX Heading, followed by printout of new S/R listing.	
4580	* MAIN SUBROUTINE YX DELETED Message(s) specifying name(s) of main S/R(s) calling the S/R that has been deleted. These main S/R(s) are also deleted.	
4599	* END OF SUBROUTINE LIBRARY STRIP	

4) Example

- <u>Problem</u>: To remove two S/R's from the ST.
  S/R Names: YPNST, YMULS .
  Order card belongs to sequence A103
- <u>Solution</u>: Punch cards as follows:


1
5
10
15
20
70
75
80
1
2
3
4
5
6

```
.SLS                                                                    A103
 SLSYPNST
 SLSYMULS
 END
```

g. .SLI Subroutine Library Insert

NOTE: Refer to the GE-130 APS Programmer's Manual for definitions of APS terms used in this Order and identified by quotation marks.

1) Purpose and Operation

This program is used to insert a S/R (which has just been assembled on WT2 by the .APS Order) into the S/R Library on the ST. The program can be subdivided into three phases:

a) Transfer the S/R from WT2 to WT1, formatting it as required (page II-6).

b) Insert the S/R into the S/R Library.

c) Update the S/R Index in the "APS Assembler" on the ST.

NOTE: When the .APS Order has been processed, the S/R is on WT2 with the following format:

- The S/R is contained in one or more blocks having fixed lengths of 695 characters each.
- Each block consists of a 5-character block number, plus five 138-character records formatted as shown below:

Characters	Contents
1 - 80	One "APS source instruction", or one of the instructions generated by an "APS source macro-instruction".
81 - 110	Not significant.

111 - 119	Reserved for error signals.
120 - 121	Length of instruction, or Length (minus 1) of areas, or Length (minus 1) of constants. (Defined by "DS, "DC" or "DP" instructions).
122 - 125	Instruction address.
126 - 127	Operation code of instruction.
128 - 129	Operation complement of instruction. ("Duplication factor" for constants and areas).
130 - 133	Address of first operand, or decimal zeroes, whichever is applicable.
134 - 137	Address of second operand, or decimal zeroes, whichever is applicable.
138	Decimal zero.

NOTE: The information in characters 120 - 138 is in unpacked binary form.

The operation of . SLI is as follows:

a) Position WT1 and WT2 after initial TM.

b) Process and transfer the S/R to WT1 with:

- Blocks formatted on WT1 as described in page II-7 except block labels not recorded.

- Each alphanumeric constant recorded in "unpacked form", and left-aligned (with blanks to the right if the actual length is less than the defined length).
- Each hexadecimal constant is translated into machine language and right-aligned (with binary zeroes to the left if the actual length is less than the defined length).
- Both alphanumeric and hexadecimal constants are recorded on WT1 as many times as specified by the "duplication factor".
- A "DS" instruction closes a block, except when it has a "duplication factor" of 0.
- An "ORG" instruction closes a block.
- The "*", "LF" and "EJEC" instructions are ignored.

c) Rewind WT1 and WT2.

d) Position the ST ahead of the S/R Index contained in the "APS Assembly Program".

e) Update the S/R Index.

f) Transfer the S/R from WT1 to the ST at the end of the S/R Library, recording a block label (figure 6, page II-6), before each block.

g) Rewind ST and WT1.

NOTE: The date recorded in the block label (field C, figure 6, page II-6) is taken by the . SLI program from:

- The Comment card, if used after the source card deck and before the End card, provided the date is specified in the operand field in the form, YNNNNN..YYDDD.
  (YNNNNN = S/R name, ƀYYDDD = Date in decimal form).

- The day's date in SRA, if the Comment card is not used, or if the date is not specified correctly.

2) Subroutine Programming Restrictions

   a) Instructions must be ordered sequentially.

   b) In "DC" ("AL1") instructions, symbolic addresses are not allowed.

   c) "DC instructions with continuation" are not allowed.

   d) In "ORG" instructions, only the "R operand" is allowed.

   e) "RORG" "ESCT" and "ESGM" instructions are not allowed.

   f) A S/R name cannot be used as an operand of an instruction unless it is called in a "JU" or "JRT" instruction. The S/R called must be in the S/R Library. (Note that the . APS program will print out a "U" message, to signal that the S/R has called another S/R).

   g) The S/R name when used in a "JU" or "JRT" instruction cannot be modified ("+" or "-").

   h) S/R's using Conversion Tables must have fixed-length names of 6 characters, beginning and ending with "Y".

3) Parameter Cards

The program uses an Order card only, formatted as follows:

Columns	Contents
1 - 4	. SLI
5 - 10	Name of S/R to be inserted.
11 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

4) Condition Checking Messages

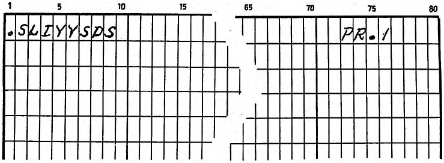
Message Number	TITLE and NOTES	Stops at Address
4601	*UNIT X: PERSISTENT ERROR (read or write)	02A6
4602	*UNIT X: OUT OF ORDER	
	If after a rewind command	01DA
	If after a read or write command	
	If during final rewind, program ends normally.	

4609	* SUBROUTINE YX NOT FOUND S/R to be inserted not in WT2, or the S/R called is not in ST.	02A6
4614	* MULTIPLE SUBROUTINE NAME YX A S/R with the same name is already in the S/R Library. The S/R is not inserted.	02A6
4638	* INVALID LABEL Label generated by APS Assembler not found in WT2.	02A6
4639	* INVALID SUBROUTINE NAME (cols. 5 - 10 of Order card) S/R name does not commence with Y.	02A6
4650	* SUBROUTINE ADDRESS WRONG The initial address of the S/R is not 0000.	02A6
4651	* SOURCE STATEMENT WRONG A programming restriction has been disregarded (see 'Restrictions') The wrong instructions are printed.	02A6
4699	* END OF SUBROUTINE LIBRARY INSERT	

5) Example

- Problem: To insert a S/R in S/R Library of ST. The Order is part of the ID Sequence, PR.1 and the S/R name is YYSDS.

- **Solution:** Punch the Order card as follows


1
5
10
15
65
70
75
80

```
.SLIYYSDS
```

```
PR.1
```

h. .SPC Subroutine Peripheral Change

1) Purpose and Operation

This program is used to change the names of one or two peripheral units within one or more S/R's in the ST. It operates in the following sequence:

a) Read and print all Detail cards (which specify the S/R's called for and the old and new peripheral unit names) converting the peripheral names to binary.

b) Position the ST at the beginning of the S/R Library, and WT1 after the Header labels.

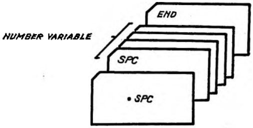
c) Transfer the contents of the S/R Library from the ST to WT1, changing the peripheral names as specified.

d) Rewind the tapes to the positions in step (b) above, and transfer the contents of WT1 to the ST.

e) Rewind both tapes.

2) Parameter Cards

The program uses an Order card, a Detail card for each S/R to be changed (sequenced in any order) and an End card. They are formatted as follows:



ORDER CARD

Columns	Contents
1 - 4	. SPC
5 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARD

1 - 4	ƀSPC
5 - 10	Name of S/R to be processed.
11 - 13	Present name of peripheral to be changed (in decimal)
14 - 16	New name of peripheral (in decimal).
17 - 19	Present name of second peripheral to be changed, if requires (in decimal).
20 - 22	New name of second peripheral (in decimal).
23 - 80	Not significant.

.SPC

END CARD

Columns	Contents
1 - 4	ƀEND
5 - 80	Not significant

3) Condition Checking Messages

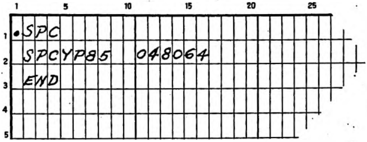
Message Number	TITLE and NOTES	Stops at Address
4701	*UNIT X: PERSISTENT ERROR (read or write)	02A6
4702	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program ends normally.	
4706	*INVALID DETAIL CARD Error in cols. 1 - 4, or End card after Order card.	02A6
4709	*SUBROUTINE YX NOT FOUND (on ST) Program continues to the end, but stops at address	02A6
4739	*INVALID SUBROUTINE NAME Name does not begin with Y. Program continues to the end, but stops at address	02A6

4740	*INVALID PERIPHERAL NAME X Name not in decimal, or greater than 255. Program continues to end, but stops at address.	02A6
4750	*SUBROUTINE YX PERIPHERAL Z NOT FOUND S/R (cols. 5 – 10 of Detail card) does not contain peripheral specified. Program continues to end, but stops at address.	02A6
4799	*END OF SUBROUTINE PERIPHERAL CHANGE	

4) Example

- Problem: To change the name of the Card Punch.
  S/R name: YP85
  Present name of Punch: 048
  New name of Punch: 064

- Solution: Punch cards as follows:



```
.SPC
 SPCYP85  048064
 END
```

i. .LTU Library Tape Update

1) Purpose and Operation

This program is used to generate a new LT or to update an existing one, as follows:

a) Insert programs from the ST into the LT, with or without modifying the File Table.

b) Remove programs from the LT.

c) Update the File Tables of existing programs.

In processing, the new LT is recorded on a separate tape, leaving the old LT intact for future reference.

The program is in three segments, operating as follows:

- First Segment (Read and Store Detail cards)

  a) Update the LT creation and purge dates, recording today's date and the same date plus three years, then printing the labels, VOL1 (or INT1) and HDR1.

  b) Read the Detail cards and store them in a table. In case of table overflow, a message is printed specifying the last card accepted. The remaining cards are not processed.

  c) If a new LT is to be created (in col. 5 of Order card) start third segment. If not, start second segment.

- Second Segment (Update existing LT)

  a) Check for PRL (see page III-128) in the table. If none, rewind ST. If present, transfer those programs that are to be inserted at the beginning of the LT from the ST to the updated LT.

b) Read first program label in the existing LT and check for the presence of corresponding Detail cards in the table. If none, transfer that program to the updated LT. If present, proceed to one of the following steps.

c) If the program name is on a PRI card (cols. 5 - 12) transfer that program to the updated LT.
Immediately following this program, insert the new program which has been transferred to the updated LT from the ST.

d) If the program name is on a PRT card, (see page III-128) transfer that program to the updated LT and update the File Table of the transferred program as required.

e) If the program name is on a PRM card (see page III-128), it is not transferred to the updated LT.

f) Subsequent program in the old LT are processed as described to the updated LT.

g) After all program labels have been processed and Detail cards remain in the table for programs not found, print out the program names under message number 4808.

Third Segment (Create new LT)

Only those programs called for by PRI cards are recorded on the new LT, in the same sequence in which the cards are read.

.LTU

2) Parameter Cards

The program uses an Order card, three types of Detail cards (described below) and an End card.

PRI (Program Insert) is used to insert a program from the ST. If the File Table of that program is to be modified, the PRI is followed by LBC cards which contain the information required for this purpose.

PRT (Program Transfer) is used to modify the File Table of an existing program in the LT. It is always followed by LBC cards.

PRM (Program Remove) is used to eliminate a program from the LT.

To create a new LT, use only PRI cards (with LBC cards if applicable). To update the existing LT, all three types if Detail cards may be used (with LBC cards, if applicable).

LBC cards are described in various Orders. The other cards are fromatted as follows:

ORDER CARD

Columns	Contents
1 - 4	.LTU
5 - 6	W1 if existing LT is WT1. W2 if existing LT is WT2. W3 if existing LT is WT3. Or the Select number of unit containing the existing LT (b̸1 to b̸8). *in col. 5 if generating a new LT.
7 - 8	Same as cols. 5 - 6, but referring to updated or new LT.
9 - 14	Serial number of tape specified in cols. 5 - 6, if checking requested. Otherwise, blank.
15 - 20	Serial number of tape specified in cols. 7 - 8, if checking requested Otherwise, blank.
21 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARDS

	PRI Card
1 - 4	ƀPRI
5 - 12	Name of program on existing LT preceding program to be inserted. Blank if program is to be inserted at the beginning of the updated LT, or if creating a new LT (in col. 5 of Order Card). If updating LT, the first PRI card only may have blanks in cols. 5 - 12).
13 - 20	Name of program to be inserted (from ST).
21 - 80	Not significant.

	PRT Card
1 - 4	ƀPRT
5 - 12	Name of program in LT containing File Table to be modified.
13 - 80	Not significant.

	PRM Card
1 - 4	ƀPRM
5 - 12	Name of program to be deleted from LT.
13 - 80	Not significant

END CARD

1 - 4	ƀEND
5 - 80	Not significant.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
4801	*UNIT X: PERSISTENT ERROR (read or write)	02A6
4802	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program ends normally.	
4806	*INVALID DETAIL CARD . Name incorrect (cols. 1 - 4) . The End card follows immediately after the Order card. . The Detail cards are not in the proper sequence. . LBC card(s) not following the PRT card.	02A6
4808	*PROGRAM TW NOT FOUND (on ST or LT) Program continues to end but stops at address	02A6

.LTU

Message Number	TITLE and NOTES	Stops at Address
4810	*UNIT X: SERIAL NUMBER Y WRONG/ SHOULD BE Z	02A6
4812	*UNIT X: HDR1 LABEL NOT FOUND (on new or updated LT)	02A6
4819	*DETAIL CARD OVERFLOW/X LAST CARD Program continues and processes only the accepted Detail cards. At the end it stops at address.	02A6
4820	*INVALID PROGRAM NAME Name in LBC card different from name in PRI or PRT card. See NOTE at end of Table.	02A6
4823	*TABLE UNIT CODE WRONG (cols. 5 - 6 and/or 7 - 8 of Order card).	02A6
4824	*MAIN UNIT CODE ERROR (col. 43 of LBC card). See NOTE at end of Table.	02A6
4825	*ALTERNATE UNIT CODE ERROR (col. 44 of LBC card). See NOTE at end of Table.	02A6
4827	*ILLEGAL DENSITY CODE (col. 51 of LBC card). See NOTE at end of Table.	02A6

Message Number	TITLE and NOTES	Stops at Address
4828	*IDENTIFICATION FILE CODE X NOT FOUND (In File Table to be updated). LBC card is not processed but program continues to end and stops at address	02A6
4829	*INVALID DATE (cols. 37 - 42 of LBC card). See NOTE at end of Table.	02A6
4834	*INVALID FILE CHECK OPTION CODE (cols. 45 - 50 of LBC card). See NOTE at end of Table.	02A6
4838	*FILE TABLE NOT FOUND Related program is recorded on LT anyway. At the end, the program stops at address	02A6
4879	*LIBRARY TAPE INDEX This message is followed by a listing of programs on new or updated LT.	
4899	*END OF LIBRARY TAPE UPDATE	

NOTE: When error is in LBC card, program checks the other LBC cards then stops at address 02A6.

4) **Example**

- **Problem:** To insert a new program in the LT.
  Follows program : INVTY.␢A
  New program name: INVTY.␢B
  Existing LT is mounted on Unit 3
  New LT to be written on WT2
  Checking of tapes not required
  File Table of new program does not require modifcation.

  Solution: Punch cards as follows:

```
.LTU 3W2
 PRIINVTY. AINVTY. B
 END
```

j. .MTG Master Tape Generate

1) Purpose and Operation

This program is used to generate a MT, taking Type 1 programs from the ST and Type 3 programs from the LT. The following Type 1 programs are always inserted:

Supervisor
.INT (Initialize)
.PRL (Program Load)
.MTS (Master Tape Stop)

To allow the system to operate in MTM (Master Tape Mode), the Supervisor, .INT, .PRL, and each Type 3 program are modified as follows:

a) In the Supervisor, Pilot 1 is altered so that, before reading an Order card, it checks the MT Control Area in SRA.

b) In .INT all references to ST are changed to MT.

c) Before loading a program from the MT, .PRL changes the jump instruction address in the Re-entry area of SRA, from Pilot 0 to Monitor. Thus, after its execution, the program transfers control to its Monitor instead of to Pilot 0.

d) On each Type 3 program, segment 00 is added to the beginning, and segment 99 is appended to the end. Segment 00 contains the BIOS routines required for that program, plus the following optional routines which are used to preset the program for execution:

- Load data in the USDA.
- Load hexadecimal 'OA' in the Core Store area available to the user program.
- Execute 'HLT' before processing the program.

Segment 99 is the Monitor, used to load and start the next program. It may perform one of five functions, depending on the needs of the programmer. These are specified in the routing code (the first character of the Monitor) by a value from 1 to 5. The format of the Monitor is shown in figure 9 , page II - 11, and its operation is described in page III - 17.

If required, the .MTG Order also can update the File Table of each Type 3 program by using LBC Detail cards.

The .MTG program operates as follows:

a) Transfer the following Type 1 programs to the MT:

- Supervisor (with Pilot 1 modified).
- .INT (with references changed to MT).
- .PRL (with jump instructions changed).
- All Type 1 programs requested on OPC Detail cards.
- MTS program.

b) Transfer the required Type 3 programs to the MT, as follows:

- Read a UPC Detail card and search the LT for the requested program.

- Generate segment 00, using a segment attached to .MTG and the parameters specified in the UPC card.

c) If LBC cards are used, update the Type 3 program File Table.

d) Incorporate the proper BIOS routines in segment 00 of each Type 3 programs, as specified in character 10 of the Type 3 program label (figure 3, page II-3). The BIOS routines are recorded in a segment of the .MTG program, which was previously preset by the .STG program.

e) Transfer the Type 3 program from the LT to the MT.

f) Generate the Monitor by using a segment of .MTG and the parameters specified in the PRE Detail card.

2) Parameter Cards

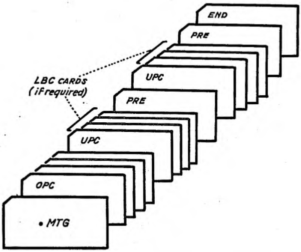
The program uses an Order card, four types of Detail cards, and an End card. The Detail cards and their functions are as follows:

OPC (Operating System Program Call) is used to insert a Type 1 program into the MT. (Note that the Supervisor, .INT, .PRL, and .MTS are inserted automatically).

UPC (User's Program Call) is used to insert a Type 3 program.

PRE (Program End) is used to preset the Monitor for each Type 3 program inserted. It follows immediately after a UPC card, or after the LBC cards used with the UPC.

.MTG



ORDER CARD

Columns	Contents
1 - 4	.MTG
5 - 6	W1 if LT is WT1. W2 if LT is WT2. W3 if LT is WT3. Or the Select number of unit containing the LT (b̸1 to b̸8).
7 - 8	Unit containing MT, identified as in cols. 5 - 6.

9 - 14	Serial Number of LT if checking is desired. Otherwise, blank.
15 - 20	Serial number of MT is checking is desired. Otherwise, blank.
21 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARDS

	OPC Card
1 - 4	ƀOPC
5 - 12	Name of Type 1 program to be inserted.
13 - 80	Not significant.

	UPC Card
1 - 4	ƀUPC
5 - 12	Name of Type 3 program to be inserted.
13 - 14	Not significant.
15	Used for segment 00 generation. H if HLT is to occur before starting program. Otherwise, blank.
16	Memory loading code used for segment 00 generation. ƀ if Core Store area available to user is to be set to hexadecimal 'OA' before starting program. * if otherwise.

.MTG

17	Used for segment 00 generation * if data to be loaded in USDA before starting program. ɓ if otherwise.
18 - 49	Data to be loaded in USDA if in col. 17.
50 - 80	Not significant.

LBC Card

Since these cards are used by other Orders as well, they are described in Appendix D. In this Order, one LBC is used for each data file, if any, of a Type 3 program if the associated File Table is to be modified.

	PRE Card
1 - 4	ɓPRE
5 - 12	Name of program to which Monitor belongs. (Same name as given in cols. 5 - 12 of preceding UPC card).
13	Monitor Routing Code: (values 1 to 5) 1. Load and start the program specified in cols. 14 - 21. 2. Load and start one program selected by setting of SWITCH 2, between the two programs specified in cols. 14 - 21 and 22 - 29.

	3. Read an Order card. 4. Execute .MTS (Master Tape Stop). 5. Load and start program specified by the program to which Monitor belongs.
14 - 21	If '1' in col. 13, name of program to be started next. If '2' in col. 13, name of program to be started with SWITCH 2 off.
22 - 29	If '2' in col. 13, name of program to be started with SWITCH 2 on.
30 - 80	Not significant.

END CARD

1 - 4	ƀEND
5 - 80	Not significant.

.MTG

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
4901	*UNIT X: PERSISTENT ERROR (read or write)	02A6
4902	*UNIT X: OUT OF ORDER	
	If after a rewind command	01DA
	If after a read or write command	02A6
	If during final rewind, program ends normally.	
4906	*INVALID DETAIL CARD	02A6
4908	*PROGRAM TX NOT FOUND Program indicated in cols. 5 - 12 of OPC or UPC card not in ST or LT.	02A6
4910	*UNIT X: SERIAL NUMBER Y WRONG/SHOULD BE Z Serial number of LT or MT not the same as that indicated in cols. 9 - 14 or 15 - 20 of Order card.	02A6
4920	*INVALID PROGRAM NAME Name indicated in LBC or PRE different from that in preceding UPC card.	02A6
4923	*TAPE UNIT CODE WRONG (Order card, cols. 5 - 6 for LT, cols. 7 - 8 for MT)	02A6
4924	* MAIN UNIT CODE WRONG (col. 43 of LBC card) see NOTE at end of Table.	02A6

4925	*ALTERNATE UNIT CODE WRONG (col. 44 of LBC card) See NOTE at end of Table	02A6
4926	*ILLEGAL DENSITY CODE (col. 51 of LBC card). See NOTE at end of Table.	02A6
4928	*FILE IDENTIFICATION CODE X NOT FOUND File indicated in cols. 13 - 18 of LBC card not in program's File Table.	02A6
4929	*INVALID DATE (cols. 37 - 42 of LBC card). See NOTE at end of Table.	02A6
4931	*USER DATA CODE WRONG (col. 17 - UPC card)	02A6
4932	*MEMORY LOADING CODE WRONG (col. 16 - UPC card)	02A6
4933	*HALT CODE WRONG (col. 15 - UPC card)	02A6
4934	*INVALID FILE CHECK OPTION CODE (cols. 45 - 50, LBC card). See NOTE at end of Table.	02A6
4937	*FILE TABLE NOT FOUND Program indicated in LBC card has no File Table. This message printed after message 4906.	

.MTG

4950	*MONITOR ROUTING CODE WRONG (col. 13 - PRE card)	02A6
4951	*MONITOR PROGRAM TABLE OVERFLOW Insertion on MT requested for more than 77 programs (the maximum number of program names that can be recorded in Field F of Monitor, see Figure 9, page II-11). However the MT is closed normally after the programs that have been inserted up to this point.	02A6
4999	*END OF MASTER TAPE GENERATE	

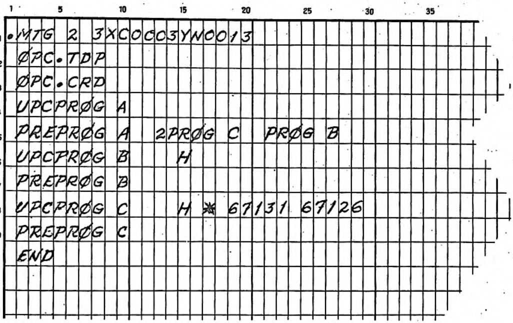
NOTE: When error is in LBC card, the program finishes checking that card, before stopping at address 02A6.

4) Example

- Problem: To record MT on tape unit 3, containing:
  Type 1 programs: .TDP and .CRD
  Type 3 programs: PROGƀA
  PROGƀB
  PROGƀC

  Preset PROGƀC to load user data in USDA.
  Precede PROGƀB and C with a HLT.
  After PROGƀA execute PROGƀB with SWITCH 2 on or PROGƀC with SWITCH 2 off.
  After PROGƀB execute PROGƀC.
  After PROGƀC read an Order card.
  LT mounted on tape unit 2 with serial number XC0003.
  MT serial number: YN0013.

Solution: Punch cards as follows:



```
.MTG 2 3XC0C03YN0013
 0PC.TDP
 0PC.CRD
 UPCPR0G A
 PREPR0G A  2PR0G C  PR0G B
 UPCPR0G B    H
 PREPR0G B
 UPCPR0G C    H * 67131 67126
 PREPR0G C
 END
```

1
5
10
15
20
25
30
35
1
2
3
4
5
6
7
8
9
10
11
12

k. . BPR Basic Program Replace

1) Purpose and Operation

This program is used to duplicate the ST, replacing one or more blocks in the Supervisor. The program operates as follows:

a) Duplicate ST up to the block to be replaced, as indicated in column 5 of BRP Detail card.

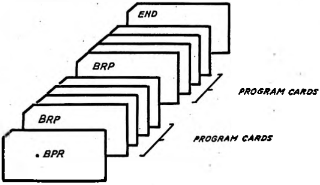
b) Insert the block contained in the Program cards following the BRP card.

c) Repeat steps a and b for all blocks to be replaced.

d) Complete ST duplication then rewind orginal and updated ST.

2) Parameter Cards

The program uses an Order card, one or more BRP Detail cards, Program cards are required, and an End card, formatted as follows:



ORDER CARD

Columns	Contents
1 - 4	. BPR
5 - 6	W1 = if new ST is WT1 W2 = if new ST is WT2 W3 = if new ST is WT3 Or Select number of tape unit containing new ST (ƀ1 to ƀ8).
7	Recording density code: K = 1600 bpi L = 800 bpi M = 556 bpi
8 - 13	Serial number of tape to be recorded, if check desired. Otherwise, blank.
14 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARD

1 - 4	bBRP (Block Replace)
5	Number of block to be substituted.

.BPR

6 - 9	Number of program cards containing the block to be substituted, if check desired. Otherwise, blank
10 - 80	Not significant.

END CARD

1 - 4	ƀEND
5 - 80	Not significant.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
5001	*UNIT X: PERSISTENT ERROR (read or write)	02A6
5002	*UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, the program ends normally.	
5006	* INVALID DETAIL CARD (cols. 1 - 4, BRP card)	02A6

5010	*UNIT X: SERIAL NUMBER X WRONG/SHOULD BE N Serial number of tape being recorded is different from that given in Order card.	02A6
5011	*CARD COUNT WRONG (cols. 6 - 9, BRP card)	02A6
5023	*TAPE UNIT CODE WRONG (cols. 5 - 6, Order card)	02A6
5027	*ILLEGAL DENSITY CODE (col. 7, Order card)	02A6
5041	*DATA CARD WRONG Program card has characters not in hexadecimal, or a program code different from first card of the block.	02A6
5044	*BLOCK LENGTH WRONG Last program card exceeds maximum block length of 4096 characters.	02A6
5047	*BLOCK NUMBER WRONG (col. 5, BRP card)	02A6
5050	*INVALID DATA CARD ADDRESS Program card with address less than first card of the block.	02A6
5099	*END OF BASIC PROGRAM REPLACE	

.BPR

4) Example

- Problem: To replace the first block of the Supervisor. Record new ST on WT2 at 556 bpi.

- Solution: Punch cards as follows:

```
.BPRW2M
 BRP1
  }(Program Cards containing new Block.)
 END
```

1. .LPD Library Tape Print and Duplicate

1) Purpose and Operation

This program permits the duplication of the LT while printing out the labels of all programs in the tape. Either duplicating or printing is optional.

LT to be duplicated/printed is WT1.
Duplicated LT (if applicable) is WT2.

The program can also be used to adapt an LT generated by the GE-115 under TOS (Tape Operating System), for operation under ETOS.

The program operates in the following sequence:

a) The tape(s) are positioned after the TM following the two Header labels.

b) The LT is duplicated and/or labels are printed.

c) When duplicating a 115-generated LT (asterisk in column 6 of Order card), the code recorded in character 10 of each program label to specify the type of GE-115 BIOS, is converted into the applicable code for ETOS (as shown in field C of figure 3, page II-3).

NOTE: In duplicating each two consecutive blocks in the same segment, they are consolidated in one block, with maximum length increased from 2048 characters to 4096 characters, on tape.

2) Parameter Cards

The program uses an Order card only, formatted as follows:

Columns	Contents
1 - 4	.LPD
5	b̸ to duplicate and print D to duplicate only P to print only
6	b̸ to duplicate GE-130 LT to generate GE-130 LT from GE-115 LT
7 - 72	Not significant
73 - 76	Optional ID Code.
77 - 80	Blank

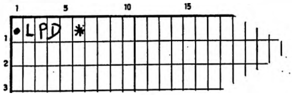
3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
5301	*UNIT X: PERSISTENT ERROR (read or write)	02A6
5302	*UNIT X: OUT OF ORDER If after rewind command If after read or write command If during final rewind, program ends normally.	 01DA 02A6
5318	* EXECUTION TYPE CODE WRONG (col. 5, . LPD card)	02A6
5350	*LIBRARY TAPE CODE WRONG (col. 6, . LPD card)	02A6
5399	*END OF LIBRARY TAPE PRINT AND DUPLICATE	

4) Example

- Problem: To generate a GE-130 LT from a GE-115 LT, to permit operation under ETOS. Print out the LT thus obtained.

- Solution: Punch the following Order card:



•LPD *

m. LTM Library Tape Merge

1) Purpose and Operation

This program is used to record on one LT either all or part of the programs contained in another LT. The programs to be recorded are added after those already contained in the LT. The LT from which the programs are to be transferred is assigned as WT1 by the .INT Order, and the other LT is assigned as WT2.
The program operates in the following sequence:

a) Position WT1 at beginning of tape and WT2 after last program.

b) Read the Detail cards (if b̸ in col. 5 of Order card), specifying the programs to be transferred from WT1 to WT2.

c) The programs specified by the Detail cards (if b̸ in col. 5 of Order card), or all the programs (if in col. 5), are transferred frol WT1 to WT2, in the same sequence in which they are recorded on WT1.

NOTE: The two LT's may have been generated by the GE-115 under TOS (Tape Operating System).

2) Parameter Cards

The program uses an Order card, one Detail card (LPI) for each program to be transferred and an End card. If entire program library is to be transferred, no Detail cards are required. They are formatted as follows:

ORDER CARD

Columns	Contents
1 - 4	.LTM
5	* if entire program library on WT1 is to be transferred to WT2. ƀ if programs to be transferred from WT1 to WT2 must be specified by Detail cards.
6 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

DETAIL CARD

	LPI (Library Program Insert)
1 - 4	ƀ LPI
5 - 12	Name of program to be transferred.
13 - 80	Not significant.

END CARD

1 - 4	ƀ END
5 - 80	Not significant.

.LTM

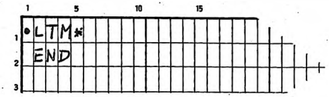
3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at address
5401	* UNIT X : PERSISTENT ERROR (read or write)	02A6
5402	* UNIT X : OUT OF ORDER If after a rewind command If after a read or write command If during final rewind, program ends normally.	 01DA 02A6
5406	* INVALID DETAIL CARD Name incorrect (cols.1-4), or Order card, with * in col.5, not followed by End card, or Order card, with b̸ in col.5, followed by End card.	02A6
5408	* PROGRAM 3X NOT FOUND (on WT1) Program continues to end but stops at address	02A6
5413	* MULTIPLE PROGRAM NAME 3X Program found on WT2 with same name as program to be inserted. Program is inserted anyway. .LTM program continues to end, then stops at address	02A6
5418	* EXECUTION TYPE CODE WRONG (col.5, LTM card)	02A6
5419	* DETAIL CARD OVERFLOW Available Core Store area cannot contain entire list of programs. Program continues to process only the accepted Detail cards. At the end it stops at address	02A6
5499	* END OF LIBRARY TAPE MERGE	

4) Example

- **Problem**: To insert all programs of LT1 after the programs on LT2.

- **Solution**: Punch the following Parameter cards:



```
•LTM*
END
```

NOTE: LT1 and LT2 were previously defined as WT1 and WT2 respectively, by the .INT Order.

n. .TPU ST Program Update

1) Purpose and Operation

This program is used to generate a new System Tape on WT2, by inserting in the current ST new and/or updated programs from another ST on WT1.

The program consists of five segments, operating as follows:

.Segment 1 (Read first Detail card and call appropriate segments)

a) Rewind WT1 and WT2 and check labels (if requested)

b) Print WT2 header labels.

c) Read first Detail card:
If END card, call Segment 2,
If OSR followed by END card, call Segment 3.
Otherwise, call Segment 4.
Block read/write area is unique and occupies 4096 Core Store locations.

.Segment 2
Duplicate current ST on WT2

.Segment 3 (Replace Operating System)
Generate new ST on WT2, by copying type 1 programs (except for Supervisor) and S/R library from WT1, Supervisor and type 3 programs from current ST.

.Segment 4 (Read and store Detail cards)

a) Read the Detail cards and store them in a table. If table overflow occurs a message is printed out specifying the last card accepted. In this case, processing can be completed by calling the Order program again and using WT2 as the current ST and the excess Detail cards.

Note that each run can process up to:
50 Detail cards using cols. 1-22, or
90 Detail cards using cols. 1-13.

b) Check Detail cards. If no errors are detected, call Segment 5, otherwise complete checking until the END card.

.Segment 5 (Update current ST)

a) Rewind ST and check labels (if requested).

b) Copy Supervisor from ST to WT2.

c) Copy, update or insert type 1 programs from ST or WT1.

d) Print labels of programs recorded on WT2 specifying their origin (either ST or WT1).

e) While recording type 1 programs on WT2, store their names in a table in Core Store, testing it each time to avoid inserting a program twice.

This table can hold up to 188 names.

f) While reading WT1, store in a table the names of the programs encountered. Thus search time for a given program on tape is reduced by executing a backward search if the program is referenced in the table, or otherwise a forward search.

This table can hold up to 188 names.

g) Before recording the S/R library rewind WT1 and insert in WT2 the other type 1 programs, if any, whose position on WT2 was not specified.

h) Insert the S/R library from WT1, if the .APS program was replaced, otherwise from ST.

i) Copy, update or insert type 3 programs, as described for type 1 programs.

j) Print the unprocessed Detail cards (see message 5941).

k) Rewind tapes.

NOTE: As a result of errors or unprocessed Detail cards, programs on WT2 are erased and further recording is inhibited, but processing continues to the end, to complete error checking.

2) Parameter Cards

The .TPU program requires an Order card, up to four types of Detail cards (described below) and an End card.

PIN (Program Insert) is used to insert a new program from WT1.

PRR (Program Replace) is used to replace a program on ST with an updated version on WT1.

PRD (Program Delete) is used to delete a program or a series of consecutive programs on ST.

OSR (Operating System Replace) is used to replace the Operating System on ST (except for the Supervisor) with an updated version on WT1.

The Parameter cards are formatted as follows:

ORDER CARD

Columns	Contents
1 - 4	.TPU
5 - 10	Serial number of current ST, if checking requested. Otherwise, blank.
11 - 16	Serial number of WT1, if checking requested. Otherwise, blank.
17 - 22	Serial number of new ST (WT2), if checking requested. Otherwise, blank.
23 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank

DETAIL CARDS

1 - 4	ƀ PIN
5 - 13	Type (1 or 3) and name of program to be transferred from WT1 to WT2.
14 - 22	Type (1,2 or 3) and name of program on ST, which must be followed by the program specified in cols. 5 - 13. If 1BASICPRG, insert program after Supervisor. If 2BASICSUB, insert program after S/R library. If all blanks, insert program at the end of the programs of the same type (see col.5); the sequence of program insertion is the same as programs are recorded on WT1.
23 - 80	Not significant

NOTES: The PIN card cannot be used to insert the Supervisor (1BASICPRG), the .APS program and the S/R library (2BASICSUB).

Two or more consecutive programs can be transferred by using several PIN cards. (See example in page III-150.18).

Columns	Contents
1 - 4	ⱥPRR
5 - 13	Type (1 or 3) and name of program on ST to be replaced by program on WT1 having same name. Location of updated program on WT2, relative to other programs, will be the same as on the original ST.
14 - 80	Not significant.

NOTES: If the program to be replaced is .APS, the S/R library (2BASICSUB) also is replaced: it is not allowed to specify explicitly the substitution of 2BASICSUB.

The example in page III-150.18 shows how to insert several consecutive programs (using PIN cards) after the program to be replaced.

Columns	Contents
1 - 4	ƀPRD
5 - 13	Type (1 or 3) and name of program not to be transferred from current ST to WT2. To delete a sequence of adjacent programs, punch type (1 or 3) and name of first program in sequence.
14 - 22	Blanks, to delete one program only. To delete a sequence of adjacent programs, punch type (1,2 or 3) and name of program following the last of the sequence. To delete a sequence of type 1 programs, starting from the one specified in columns 5-13 up to the beginning of the S/R library punch "2BASICSUB". To delete a sequence of type 3 programs, from the one specified in columns 5-13 up to the end of the tape, punch nine exclamation points (!).

NOTES: The PRD card cannot be used to delete the Supervisor (1BASICPRG), the .APS program or the S/R library (2BASICSUB).

One or more consecutive programs can be transferred from WT1 and inserted after the program specified in cols. 14-22, by using PIN cards.A PRD and a PIN card can be used to change the originalposition of a program.

Columns	Contents
1 – 4	ƀ OSR
5 – 80	Not significant

NOTES: The OSR card is used to copy to WT2: all type 1 programs (not including the Supervisor) and the S/R library from WT1, Supervisor and type 3 programs from current ST.

END CARD

Columns	Contents
1 – 4	ƀ END
5 – 80	Not significant

NOTE : If END card immediately follows .TPU card, ST is only duplicated on WT2.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at address
5901	* UNIT X : PERSISTENT ERROR (read or write)	02A6
5902	* UNIT X : OUT OF ORDER	
	If after rewind command	01DA
	If after read or write command	02A6
	If during final rewind, program ends normally	
5905	* PROCEEDING IMPOSSIBLE	02A6
5906	* INVALID DETAIL CARD Name incorrect (cols. 1 - 4): Program checks the other Detail cards. At the end, message 5905 is printed out and program stops at address	02A6
5908	* PROGRAM TX NOT FOUND (on ST or WT1)	02A6
5910	* UNIT X : SERIAL NUMBER X WRONG/ SHOULD BE Y	02A6
5913	* MULTIPLE PROGRAM NAME TX Program to be inserted is already present on ST	
5919	* DETAIL CARD OVERFLOW/X LAST CARD Program continues and processes only the accepted Detail cards, then stops at address	02A6

5921	* ILLEGAL PROGRAM TYPE (cols. 5 and/or 14 of Detail cards) Program checks the other Detail cards. At the end, message 5905 is printed out and program stops at address	02A6
5941	* DATA CARD WRONG Followed by printout of incorrect card.	02A6
5979	* SYSTEM TAPE INDEX Followed by list of programs on new System Tape. Either of the following specifications may appear. - DUPLICATE duplication from ST to WT2 was requested. - DUPLICATE: PROGRAMS TYPE 1 AND 2 FROM WT1 AND PROGRAMS TYPE 3 FROM ST If replacement of Operating System was requested.	
5999	* END OF ST PROGRAM UPDATE	

4) Example

- **Problem:** To generate a new System Tape from the following input tapes:

Current ST

1BASICPRG
1A
1B
1.APS
1C
1D
1L
1M
2BASICSUB
3P
3Q
3R
3T
3Y
!!!!!!!!!

WT1

1BASICPRG
1I
1B
1.APS
1H
1E
1N
1K
1L
1F
1J
2BASICSUB
3S
30
3W
3R
3V
3P
3U
3Z
!!!!!!!!!

Delete programs 1A, 1C, 1D, 3P and 3Q.
Replace programs 1.APS, 1L and 3R.

Insert the following programs:
- 1E, after Supervisor
- 1F, 1I, 1N, sequentially after program 1L
- 1K and 1H, after all type 1 programs
- 30, after S/R library.

Invert positions of 1.APS and 1B programs.
Checking of tape serial numbers not required.

- <u>Solution:</u> Punch cards as follows:

```
.TPU
 PRD1A
 PRD1C       1L
 PRD3P       3R
 PRR1.APS
 PRR1L
 PRR3R
 PIN1E       1BASICPRG
 PIN1F       1L
 PIN1I       1F
 PIN1N       1I
 PIN1K
 PIN1H
 PIN30       2BASICSUB
 PRD1B
 PIN1B       1.APS
 END
```

NOTE: Sequence of Detail cards is irrelevant, except when the position of a program must be changed. In this case the sequence of Detail cards must conform to the logical sequence of operations (i.e. first a PRD, then a PIN card), otherwise message 5913 would be printed out.

The following listing shows the sequence of programs on the new ST (WT2) and the origin of each:

WT2	Origin
1BASICPRG	ST
1E	WT1
1.APS	WT1
1B	WT1
1L	WT1
1F	WT1
1I	WT1
1N	WT1
1M	ST
1K	WT1
1H	WT1
2BASICSUB	WT1
30	WT1
3R	WT1
3T	ST
3Y	ST
!!!!!!!!!	

5. OTHER ORDERS

a. .CMT Comment

1) Purpose and Operation

This Order is used to print a one-line comment contained in the Order card.

2) Parameter Cards

The program uses an Order card only, formatted as follows:

Columns	Contents
1 - 4	.CMT
5 - 72	Comments.
73 - 76	Optional ID Code.
77 - 80	Blank.

b. .HLT Halt

1) Purpose and Operation

This Order is used to stop processing. When the Order is used, Pilot 1 stops at address 01D6. Pressing START restarts Pilot 0 and another Order card is read.

2) Parameter Cards

The Order uses an Order card only, formatted as follows:

Columns	Contents
1 - 4	.HLT
5 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

c. .MTS Master Tape Stop

1) Purpose and Operation

This Order causes a HLT at address 01D6 and rewinds the MT. Pressing START restarts Pilot 0.

2) Parameter Cards

The Order uses an Order card only, formatted as follows:

Columns	Contents
1 - 4	.MTS
5 - 72	Not significant
73 - 76	Optional ID Code.
77 - 80	Blank.

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
6002	UNIT X: OUT OF ORDER Followed by message 6084	
6084	MASTER TAPE STOP Press START to restart Pilot 0.	01D6

d. .RST Restart

1) Purpose and Operation

This Order is used to resume a given program from a predetermined point. The restart point (Check point) is requested by the CHKP macro-instruction and recorded on any standard file only on 9-track tape. It is formatted as follows:

a) Tape Mark.

b) Label containing Check point sequence number and restart address.

c) Two blocks containing the BIOS positions to be restored.

d) Block holding Core Store contents.

e) Tape Mark.

The operator selects the Check Point sequence number to be used. The .RST Order restores the previous Core Store contents and resets the tapes to their previous positions; according to the Check Point sequence number. It then restarts the program after the CHKP instruction.

The program operates in the following sequence:

a) Search for specified restart point.

b) Check and re-position files as indicated in File Table. The check is made only on standard files that were open when the Check Point was recorded.

c) Reload previous Core Store contents from the program initial address. (It also reloads that part of SRA used by the program except for the read and print areas, which are occupied by a part of .RST program).

d) Jump to User routine, if any, to check non-standard files, or to execute any other operation before the program is resumed.

e) Reposition ST (or MT) on program to be resumed.

f) Resume processing after the CHKP instruction.

2) Parameter Cards

The program uses an Order card only, formatted as follows:

Columns	Contents
1 - 4	.RST
5 - 8	Check Point sequence number (decimal).
9	Select number of tape unit on which tape containing Check Point is mounted.
10	Parity Check Code O if even P if odd
11	Recording density code K = 1600 bpi L = 800 bpi

12 - 17	Serial number of tape containing Check Point.
18 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

3) Condition Checking Message

Message Number	TITLE and NOTES	Stops at Address
6101	UNIT X: PERSISTENT ERROR (read)	02A6
6102	UNIT X: OUT OF ORDER	
	If after rewind command	01DA
	If after read command	02A6
	If during final rewind, program ends normally.	
6108	PROGRAM TX NOT FOUND Program to be resumed not found on MT or ST.	02A6
6110	UNIT X: SERIAL NUMBER Y WRONG/SHOULD BE Z Tape mounted on unit indicated in col. 9 has different serial number than that indicated in cols. 12 - 17.	02A6

6117	UNIT X: NON STANDARD FILE File where Check point recorded is non-standard.	02A6
6123	TAPE UNIT CODE WRONG (col. 9)	02A6
6126	ILLEGAL PARITY CHECK CODE (col. 10)	02A6
6127	ILLEGAL DENSITY CODE (col. 11)	02A6
6150	UNIT X: CHKP N NOT FOUND (N = Sequence Number)	02A6
6151	CHKP - BLOCK INCORRECT Block d, page III-154, does not cover entire Core Store from program initial address to Core Store final address.	02A6
6152	UNIT X: POSITIONING ERROR (refers to tapes used by program to be restarted)	02A6
6199	END OF RESTART/N Indicates Check Point Sequence number.	

.RST

4) Example

- Problem: To resume a program from CHKP number 0033, with following data pertaining to Check Point Tape:

Unit number:	3
Parity:	odd
Recording Density:	800 bpi
Serial number:	0A7913
Recording mode:	9-track

- Solution: Punch the Order card as follows:

```
1   5    10   15   20   25
.RSTC0333PL0A79139
```

e. .CDC Cards Duplicate

1) Purpose and Operation

This Order is used to duplicate and print out (if requested) a card deck. The program also prints out any card that has been punched incorrectly, but each of these cards is preceded by two asterisks.

2) Parameter Cards

The program uses an Order card and an End card, formatted as follows:

ORDER CARD

Columns	Contents
1 - 4	.CDC
5	b̸ = Duplicate and Print. D = Duplicate only.
6 - 72	Not significant.
73 - 76	Optional ID Code.
77 - 80	Blank.

.CDC

END CARD

Columns	Contents
1 - 4	ƀ...
5 - 80	Not significant

3) Condition Checking Messages

Message Number	TITLE and NOTES	Stops at Address
5202	* UNIT X: OUT OF ORDER (during final rewind) Program ends normally.	
5207	* PUNCH ERROR ON CARD Incorrect card(s), identified by two asterisks, are printed out. The program proceeds to its end, then stops at address	02A6
5215	* PUNCH NOT CONNECTED	02A6
5218	* EXECUTION TYPE CODE WRONG (col. 5, .CDC card)	02A6
5299	* END OF CARDS DUPLICATE	

PART IV - EXAMPLES

A. ASSEMBLE AND DEBUG APS PROGRAM

1. Problem

a. Assemble the source program.

b. Record a Sample File on tape, for debugging.

c. Print out Core Store and the Output tape to check the program for correct execution.

2. Solution

a. The Flow Chart in figure 13 illustrates the ETOS operations required.

b. The sequence and formats of the Parameter cards to be used, are shown on the Card Formatting Sheet.

3. Notes

a. The .APS, .API, .SGE and .PRL Order cards should have the same ID Code so that each of the .API, .SGE, and .PRL programs will only be started if the previous program was correctly executed. The final check operations must be accomplished in any case, so the .MDP and .TDP cards must have an ID Code blank.

b. After the .APS Order card, insert the source program card deck.

c. After the End card of the .PRL Order, insert the file on cards, if any.

d. Use a Pilot 0 card to initialize ETOS.

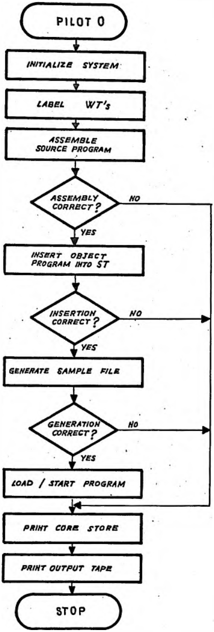


FIGURE 13 - APS PROGRAM ASSEMBLY / DEBUG FLOWCHART

```
•INT 68131*     1+     2+
•TLW
 TLW9 1PLF
 TLW9 2PLF
 END
•APS  TESTPROG                                                          P001
•API3TESTPROG                                                           P001
•SGE 1PL 3                                                              P001
 SBL00001
 BLK71ABCDEFGHIJKLMNOPQ                                               ZZZZZZZ
 BLK7199999999988888888                                               ZZZZZZZ
 EBL
 ENDF
•PRLTESTPROG                                                            P001
 LBCTESTPROGFILE01                          NNNNNNL
 LBCTESTPROGFILE02                          NNNNNNL
 END
•MDP
•TDP 2PL
•HLT
```

B. INSERT UPDATED S/R INTO ST

1. Problem

Insert a new version of S/R YAAA into the S/R Library of the ST, by replacing the old version. The S/R is written in GE-120/130 APS.

2. Solution

The Flow Chart in figure 14 illustrates the ETOS operations required.

3. Notes

a. .SLS, .APS and .SLI Order cards should have the same ID Code so that the insertion of the new S/R is done if the indicated operations were correctly executed.

b. Insert the S/R card deck after the .APS Order card.

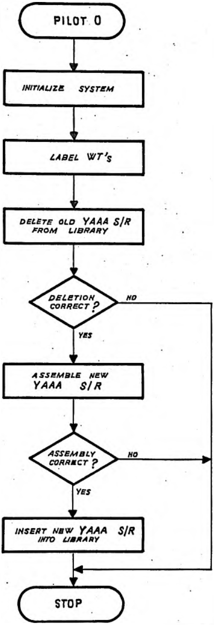


FIGURE 14 - REPLACE SUBROUTINE FLOWCHART

```
•INT 68251*     1+     2+
•TLW
 TLW9W1PLF
 TLW9W2PLF
 END
•SLS                                                                    P002
 SLSYAAA
 END
•APS 1YAAA                                                              P002
•SLIYAAA                                                                P002
•HLT
```

PART V - APPENDIXES

A. GLOSSARY OF TERMS

TERM	DEFINITION
Binary Zero	The octet, character, or field is filled with zero bits.
Block	An item of information normally treated as an entity by hardware (physical record). On magnetic tapes, a series of characters that are read or recorded during a single data transfer operation. Each block is followed by an inter-block (or inter-record) gap set to binary zero.
Block Sequence Number	Decimal number (five digits in length) specifying the sequential order of a data block within a tape. Recorded in the first five characters of a numbered block. If not present, the block is called un-numbered.
Blocks, Normalized	Blocks of same length.
Decimal Form	Representation of the digits 0 to 9.
Decimal Zero	Representation of a zero in the numeric part (right quartet) of a character or octet. When used in one octet, the left quartet will contain the hexa. value of 4.
End-of-Volume (EOV1)	See "Tape Trailer Label".
End-of-File (EOF1)	See "File Trailer Label".
File Code	See "File Identification Code".

File Creation Date	Date File was generated, recorded in File Header Label.
File Header Label	An 80-character block of data identifying the beginning of a Standard File. Follows the Tape Header Label.
File Identification Code	Alphameric File identifier used within a program. (6 characters long).
File Name	Alphanumeric File identifier recorded in the 12-character field of File Header Label
File Purge Date	Date indicating the day after which the File can be deleted or updated. Recorded in File Header Label.
File Retention Period	Length of time during which the File must not be modified (expressed in number of days).
File Trailer Label (EOF1)	An 80-character block of data identifying the end of a Standard Data File, and end of a tape. It contains the quantity of blocks in the File.
File, Standard	File containing standard blocks, standard Header (VOL1 or INT1, and HDR1) and Trailer Labels (EOV1 or EOF1) and Tape Marks (TM's)
Hexa.	Abbreviation for hexadecimal.
Hexadecimal	. Conventional notation for designating the 16 possible bit configurations of a quartet. Hexadecimal values are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F. . This notation can be used in card punching to introduce in Core Store any quartet configuration: one hexa. character per column, two hexa. characters per octet. This is commonly referred to as "punching in hexadecimal".

	. Alternatively, a graphic character can be used (in place of a hexa. character) having a right quartet configuration in Core Store, equal to the desired configuration. These cards are said to be punched in "pure hexadecimal". (See GE-100 Graphic Character Set in GE-130 Reference Manual, Appendix B.).
Illegal	Refers to an item of information which is not accepted as a valid representation by the machine design or by a specific routine or program.
Invalid	See "Illegal".
Memory Trace Routine	Causes the printout of specified Core Store areas at specified points (Trace points) in a program.
Octet	The basic unit of information in the GE-100 Systems, composed of eigth binary digits (bits).
Packed Mode (also "Form")	Generally denotes the packing of two decimal digits in a single octet.
Parity bit	A check bit that indicates whether the total number of "1" bits in a character or octet (including the parity bit) is odd or even. If the tctal number of "1" bits is always even, the system uses even parity; if odd, it uses odd parity.
Path Trace Routine	Causes the printout of all jump instructions that have occurred in a program.
Program Card	A punched card containing program instructions.
Program Code	See "Program Identification Code".

Program Identification Code	Alphameric program identifier (four characters in length) specified in a Program card.
Program Label	Block of data (28 characters in length) preceding the program it describes.
Program Name	Alphanumeric program identifier (eight characters in length) recorded in Program Label.
Program Tape	See "Tape, Program".
Quartet	Half an octet
Record	Groups of characters normally treated as an entity by the program (Logical Record). One or more Records in a Block.
Reel	See "Tape".
Select Number	See "Tape Unit Select Number".
S/N	Abbreviation for "Serial Number".
Tape Header Label (Volume or Reel) VOL1 for 9-track tape, INT1 for 7-track tape)	An 80-character block of data indicating the beginning of a standard tape.
Tape Mark (TM)	A special one-character block recognized by hardware, and used as a program separator and at both ends of a standard tape.

Tape, Program	Standard Tape containing programs (ST, MT or LT).
Tape Recording Density	The number of 6- or 8-bit characters that can be stored in one inch of tape.
Tape Sequence Number (Volume or Reel)	Sequence number of tape within a File. Four digits long.
Tape Serial Number (Volume or Reel)	Alphameric tape identifier recorded in Tape Header Label (VOL1 on INT1). Six characters long.
Tape, Standard	See "File, Standard".
Tape Trailer Label (Volume or Reel) (for Data Files)	An 80-character block of data indicating the end of a standard tape. Designated by name, "EOV1". It contains the quantity of blocks in the volume.
Tape Trailer Label (Volume or Reel) (for Program Files)	A 28-character block of data indicating the end of a standard program tape (ST, LT or MT). The first nine characters are exclamation marks (!)
Tape Unit Controller Name	Controller identifier contained in Tape Unit Name. Fixed value, depending on peripheral configuration.
Tape Unit Name	Used by PER or PERI Instructions to address a tape unit. Fixed one-octet value, depending on peripheral configuration.

Tape Unit Select Number	Number assigned by the operator to a tape unit.
Trace Routine	See "Path Trace Routine" and "Memory Trace Routine".
Unpacked Mode (also "Form")	Denotes the use of the right quartet only of an octet to store a decimal digit. It may also apply to binary numbers.
Volume	See "Tape"
Work Area	Core Store area used for temporary storage of one Record before and after processing.

B. PREREQUISITE AND RELATED READING

The following material is supplied for the GE-120/130 Systems:

Codes	Title	Catalog Numbers	
		GEISI	BGE
P-A	APS Programmer's and Operator's Manual	1079	32.30.107 A
R-A	COBOL Manual	1224	32.30.202 A
R-A	ETOS Operator's Manual	1223	32.53.505 A
R-A	FORTRAN Manual	To be assigned	To be assigned
P-A	GE-130 Reference Manual	0992	33.20.001 A
P-A	GE-120 Reference Manual	1252	32.20.002 A
R-A	Single-Capstan Tape Subsystem Manual	To be assigned	To be assigned
R-A	Subroutines, Vol. 1	1225	32.31.104 A
R-A	Subroutines, Vol. 2	1246	32.31.105 A

Codes: P = Prerequisite Reading
R = Related Reading
A = Analysts, Programmers

C. SYSTEM RESERVED AREA (SRA) CHART

CORE STORE ADDRESSES				Length in Octets	CONTENTS
From (Decimal)	To (Decimal)	From (Hexadecimal)	To (Hexadecimal)		
0	39	0000	0027	40	Pilot 0
40	45	0028	002D	6	Day's Date, expressed in the form ƀYYDDD. This date is loaded by the .INT Order.
46	49	002E	0031	4	Check point counter. It is increased by 1 by any user program each time the instruction CHKP is executed.
	50		0032	1	MT Control area. Used in starting a program in MTM.
	51		0033	1	ID Sequence Validity Switch
52	83	0034	0053	32	User Special Data area (USDA). Loaded by .INT,.PRL, or .MTG Orders.
84	87	0054	0057	4	ID Sequence Code area
88	95	0058	005F	8	Program Name area
96	97	0060	0061	2	File Table Address area. Set to binary zero if program has no File Table.
98	99	0062	0063	2	Address of first Store location after Supervisor (service information for ETOS).

100	103	0064	0067	4	System configuration information
104	107	0068	006B	4	Jump instruction to BIOS Tape routine for command and validation.
108	111	006C	006F	4	Jump instruction to GE-115 BIOS Read/Print routine.
112	115	0070	0073	4	Jump instruction to Segment Load routine.
116	119	0074	0077	4	Jump instruction for restarting the Tape Read routine after a normal call, with no change in Re-entry or in the parameters. Instruction used for continuous tape reading.
120	123	0078	007B	4	Not significant.
124	125	007C	007D	2	Address of first location after SRA, available to user programs. Variable according to system confirguration.
	126		007E	1	Core Store capacity (in binary), expressed in multiples of 256.
	127		007F	1	Character H (for Hollerith card code). Character B (for Bull card code).

	128		0080	1	ST/MT unit Select number.
129	131	0081	0083	3	WT unit Select numbers. 1st octet, for WT1 2nd octet, for WT2 3rd octet, for WT3, if applicable, otherwise set to FF.
	132		0084	1	Not significant.
	133		0085	1	To be assigned.
134	213	0086	00D5	80	Card Read area.
214	255	00D6	00FF	42	GE-130 Service and Register area.
256	447	0100	01BF	192	Read/Print routine sub-area. (if Hollerith code) Area reserved for transcoding routine (if Bull code).
448	455	01C0	01C7	8	Area containing tape unit names (00 to 07).
456	457	01C8	01C9	2	HLT instruction indicating persistent read or write error. This Halt occurs after five unsuccessful read/write attempts. To make five more attempts, press START.
458	461	01CA	01CD	4	Jump instruction to the tape routine, for making further read/write attempts.

462	463	01CE	01CF	2	HLT instruction indicating tape unit out of order.
464	467	01D0	01D3	4	Jump instruction to previous HLT.
468	469	01D4	01D5	2	HLT instruction indicating block number error.
470	473	01D6	01D9	4	Jump instruction to the S/R which searches forward for the correct block, by reading the next block.
474	475	01DA	01DB	2	HLT instruction indicating tape unit out of order, after a rewind command, during a Type 1 program execution. The operator may restart the unit by pressing OPERATE (on tape unit) and START (on CPU Console).
476	479	01DC	01DF	4	Jump instruction to the S/R which attempts the rewind operation again. In case of persistent out-of-order condition, the program stops at address 02A6.
489	511	01E0	01FF	32	Area reserved for EIOS HLT instructions.
512	513	0200	0201	2	Printer Controller area

514	673	0202	02A1	160	Print area
674	677	02A2	02A5	4	MVI instruction, setting to NO (hexa. 01) the ID Switch (hexa. address 0033).
678	679	02A6	02A7	2	HLT instruction indicating that the ID Switch was set to NO. Press START, to restart Pilot 0.
680	687	02A8	02AF	8	PER instructions for resetting recording density and parity.
688	691	02B0	02B3	4	Re-entry jump area: Jump instruction either to Pilot 0 or to Monitor.
692	695	02B4	02B7	4	Monitor Control area
696	699	02B8	02BB	4	DELTA fields used by PER instructions for setting tape density and parity.
700	719	02BC	02CF	20	Service area
720	723	02D0	02D3	4	Jump instruction to GE-120/130 BIOS (for execution).
724	727	02D4	02D7	4	Jump instruction to GE-120/130 BIOS (for validation).
728	731	02D8	02D8	4	Jump instruction to GE-120/130 BIOS. card read routine (for execution)

732	735	02DC	02DF	4	Jump instruction to GE-120/130 BIOS card read routine (for validation).
736	739	02E0	02E3	4	Jump instruction to GE-120/130 BIOS print routine (for execution).
740	743	02E4	02E7	4	Jump instruction to GE-120/130 BIOS print routine (for validation).
744	751	02E8	02EF	8	Service area
752		02F0	See Note		BIOS reserved area

NOTE: The final address of the BIOS reserved area is depending upon the model of Card Reader, Printer, and Tape Units being used.

D. FILE TABLES AND LBC CARDS

LBC cards are used to update the File Tables associated with the Data Files required by the user programs. This Appendix describes the LBC cards and the File Tables.

1. General Concepts

Each Data File on tape, before being processed, must be 'opened' by the macro-instruction 'OPEN' and 'NEXT', in the APS source program (see APS Manual).

File opening consists of various checks on the File Label, such as:

- File Name and Serial Number
- File Creation and Purge dates
- Volume sequence in multi-volume File.

These checks are performed by ETOS using a File Table and LBC cards, where applicable.

The File Table contains any relevant File information. LBC cards may be used to update the File Table and to specify other checks in addition to those which are automatically carried out.
The File Table is generated by the APS Assembly Program, by using the information specified in the instruction 'FILE' in the APS source program, and by reserving some areas which are to receive the parameters specified on the LBC cards.

LBC cards are used in three cases:

- Insert the program into the LT (.LTU Order)

- Insert the program into the MT (. MTG Order).
- Program load and start up (. PRL Order).

Refer to the individual Orders for sequencing information.

2. File Table Format

A File Table is designed to receive all essential information pertaining to the Data Files of a program.

94 octets are reserved in the File Table for each File, formatted as follows:

Characters	Contents
1	Type of File: hexa. OO, if File consists of Fixed Length records (FL File), or of Unblocked records (ie. single-record blocks) (UN File) hexa. 01, if File consists of Variable Length records (VL File)
2	Binary 94 (length of File Table)
3 - 6	HDR1
7 - 23	File Name, read from LBC card.
24 - 29	File Serial Number, read from LBC card.
30 - 33	Volume Sequence Number in multi-volumes File.
34 - 37	0001 (File Sequence Number)

38 - 43	Serial Number of first volume in File.
44 - 49	File Creation Date, read from LBC card.
50 - 55	File retention period, read from the RP parameter in the 'FILE' instruction of the Source Program. If RP parameter is absent, all decimal zeroes. When the program is started, the File Purge date is computed and recorded in this field.
56	Service character
57 - 62	File Identification Code, read from the 'FILE' instruction. After the File set-up phase this field will be used as the block counter.
63 - 68	File check options, corresponding to colums 45 thru 50 in LBC card. This field is initially set to YYYYNY by the APS Assembler.
69	Select Number of Alternate tape unit (see col. 44 in LBC card).
70 -	Reserved for tape routine parameters.
86	Code specifying the File format: hexa. 00, if Standard File with numbered blocks 01, if Standard File with unnumbered blocks 80, if Non-standard File, with numbered blocks 81, if Non-standard File, with unnumbered blocks
87 - 90	Work area information.
91 - 94	Service field.

3. LBC Card Format

Each LBC card contains the parameters referring to a given File. The card is formatted as follows:

<table>
<tr><th>Columns</th><th>Contents</th></tr>
<tr><td>1 - 4</td><td>b̸LBC (Label Control).</td></tr>
<tr><td>5 - 12</td><td>Name of Program using File.</td></tr>
<tr><td>13 - 18</td><td>File Identification Code (used by the program in the File generating and processing instructions).</td></tr>
<tr><td>19 - 30</td><td>File Name, recorded on HDR1.</td></tr>
<tr><td>31 - 36</td><td>File Serial Number.
Blanks if N in col. 46.</td></tr>
<tr><td>37 - 42</td><td>File Creation Date (b̸YYDDD).
Blanks if N in col. 48.
This date may also be read from the USDA and the Date Area in SRA by leaving blanks in cols. 37 - 41 and punching a number from 0 to 5 in col. 42, chosen as follows:
<table>
<tr><th></th><th colspan="2">Date read from SRA</th></tr>
<tr><th>col. 42</th><th>(Hexadecimal)</th><th>(Decimal)</th></tr>
<tr><td>0</td><td>0028 - 002D</td><td>40 - 45</td></tr>
<tr><td>1</td><td>0034 - 0039</td><td>52 - 57</td></tr>
<tr><td>2</td><td>003A - 003F</td><td>58 - 63</td></tr>
<tr><td>3</td><td>0040 - 0045</td><td>64 - 69</td></tr>
<tr><td>4</td><td>0046 - 004B</td><td>70 - 75</td></tr>
<tr><td>5</td><td>004C - 0051</td><td>76 - 81</td></tr>
</table>
(See Note 1)</td></tr>
</table>

43	Select number of Main tape unit containing the entire File or its first volume (see Note 2).
44	Select number of Alternate tape unit, if multi-volume File is to be read sequentially from both Main and Alternate units. Select number of Main tape unit (see col. 43) if multi-volume File is to be read entirely from that unit. Blank if File is single-volume (see Note 2).
45	Y, if File Name is to be checked. N, otherwise.
46	Y, if File Serial Number is to be checked. N, otherwise.
47	Y, if volume sequence in multi-volume File is to be checked. N, otherwise.
48	Y, if File Creation Date is to be checked. N, otherwise.
49	Y, to disable all operations on the File. N, otherwise This option applies only to input files.
50	Y, if printout of HDR1 label is requested when 'OPEN' command is issued by the user program. If HDR1 label check fails, a part of the File Table is also printed. N, if HDR1 label is not to be printed.

Columns	Contents
51	File recording density: K = 1600 bpi L = 800 bpi M = 556 bpi N = 200 bpi
52 - 80	Not significant.

NOTES:

1) The date information can be recorded in the Date Area and in USDA, of SRA by the .INT, .PRL, or .MTG Orders.

2) The specification of the Select Number of Main tape unit (col. 43) and of Alternate tape unit (col. 44) is optional, because it always appears in the FILE instruction in the source program. However, when this specification appears on the LBC cards, it overrides that in the FILE instructions.

4. Processing LBC Cards

The Orders (.LTU, MTG and .PRL) that process LBC cards, carry out the following operations:

a. Check the Program Name (LBC card, cols. 5 - 12).

b. Search the File Table (of the File designated in LBC card, cols. 13 - 18) for the specified information.

c. Move File Name (LBC, cols. 19 - 30) to File Table.

d. Move File Serial Number (LBC, cols. 31 - 36) to File Table.

e. Check the date (LBC, cols. 37 - 42).

f. Move the date to the File Table.
If the date is specified in LBC, col. 42, the .LTU and .MTG Orders transfer this code to the File Table. The actual date will be read from the Date Area or USDA to the File Table, only by the .PRL Order, or by the Monitor.

g. Compute the File Purge date (.PRL or Monitor only).

h. Check the Main tape unit Select number (LBC, col. 43) for validity. This number should be 1 to 8.

i. Move Main tape unit Select number to File Table.

j. Check the Alternate tape unit Select number (LBC, col. 44) for validity. This number should be 1 to 8.

k. Move Alternate tape unit Select number to File Table.

l. Check the options specified in LBC, cols. 45 - 50 for validity.

m. Move check options (LBC, cols. 45 - 50) to File Table.

n. Check File recording density code (LBC, col. 51).

o. Move recording density information to File Table.

NOTE: The File information that is not affected by the LBC parameters is not changed.

E. SYSTEM TAPE GENERATION

1. General Concepts

Every user's system requires a specific (customized) ST for the system configuration in which ETOS and user programs are to be executed.

Generation of a customized ST always requires a Sample ST and the .STG Order. In addition, an STG Loader is required when the Sample ST does not conform to the system configuration used for ST generation (the Supervisor is not adapted to that configuration).

The STG Loader is therefore designed to:

- Adapt the Supervisor to the existing configuration.
- Load and start the .STG program, as described in the .STG Order.

2. STG Loader

a. Peripheral Unit configuration

One Card Reader or Card Reader/Punch
One Printer
Three Tape units (unit 8 for Sample ST, unit 1 for new ST and unit 2 for a WT).

b. Operation

The STG Loader operates as follows:

1) Store Segment Load routine and tape routine (which are part of the STG Loader).

2) Transfer the Read/Print Routines from .STG program (on Sample ST) to Core Store, adapting them to the system configuration as specified by the STG Loader Parameter card.

3) The jump instructions to the tape routine and the tape error HLT instructions are preset.

4) Assign select numbers to the tape units.

5) Label the new ST and the WT.

6) Load and start .STG program

c. Parameter Cards

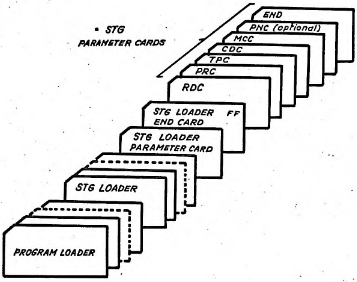
The STG Loader is contained in a card deck, which is read into Core Store together with the following cards:

1) Program Loader deck, for loading the STG Loader.

2) STG Loader Parameter card, containing system configuration information.

3) .STG Order Detail cards.

The STG Loader deck and the other cards must be introduced in the Card Reader in the following sequence:



The STG Loader Parameter card is formatted as follows:

Columns	Contents
1 - 4	2382 (card name)
5 - 6	06 (fixed)
7 - 10	0200 (fixed)

Columns	Contents
11 - 12	Model of Card Reader; b̸1= CRZ100, CRZ111, or CRZ120 b̸2= CRP100
13 - 14	Card Reader Name (in hexadecimal). Blank if CRZ100, CRZ111, or CRZ120.
15 - 16	Model of Printer: b̸1= PRT100 or PRT110 b̸2= PRT120 or PRT130 b̸3= PRT105 b̸4= PRT118
17 - 18	Printer Name (in hexadecimal). Blank if PRT100 or PRT110.
19 - 20	Blank (for Hollerith code). Any other character (for Bull code).
21 - 22	Specify recording density for new ST: A2= 1600 bpi A3= 800 bpi A4= 556 bpi
23 - 80	Not significant

c. Condition Checks and HLT Instructions

HLT Address	Description
01C8	Tape unit persistent error (read or write)
01CE	Tape unit out of order
0206	Invalid Parameter card (Error in cols. 11-12, 15-16 or 21-22)
020C	.STG Program not found on sample ST
0212	End of Program, after .STG program listing. To start .STG program, press START.

F. TAPE UNITS REQUIRED FOR INDIVIDUAL ORDERS

	ST/MT	WORK TAPES	
		Implicit in Order	Specified by Order
Group 1. a. .INT	ST or MT	None	WT1+WT2+WT3 (opt.)
b. .TLW	ST or MT	None	quantity as required
Group 2. a. .APS	ST	WT1+WT2	None
b. .COB	ST	WT1+WT2+WT3	Other tape, if source program on tape.
c. .FOR	ST	WT1+WT2+WT3	None
d. .SPG	ST	WT1+WT2	None
e. .API	ST	WT2	None
f. .PRL	ST or MT	None	None
g. .APP	ST	WT2, if program on WT2	None
h. .SBP	ST	WT1+WT2 if S/R on WT2	None
Group 3. a. .TDC	ST or MT	None	Two WT's. One WT, if duplicating ST or MT
b. .DPG	ST	WT1	None
c. .SGE	ST or MT	None	WT
d. .TDP	ST or MT	None	WT (none if printing ST or MT)
e. .CRD	ST or MT	None	None
f. .MDP	ST or MT	None	None
g. .PCC	ST	WT2, if program on WT2	None
h. .PSC	ST	WT1+WT2 if S/R on WT2	None
i. .PSI	ST	WT1+WT2	None
j. .TCK	ST	WT1+WT2	None

Group 4.	a.	.STG	ST	WT1+WT2	None
	b.	.SPD	ST or MT	None	WT (none if printout only)
	c.	.STC	ST	None	None
	d.	.STU	ST	WT1	None
	e.	.STS	ST	WT1	None
	f.	.SLS	ST	WT1	None
	g.	.SLI	ST	WT1+WT2	None
	h.	.SPC	ST	WT1	None
	i.	.LTU	ST	None	WT for new LT WT for current LT, if used.
	j.	.MTG	ST	None	WT for LT WT for MT
	k.	.BPR	ST	None	WT
	l.	.LPD	ST or MT	WT1+WT2 WT1, if printout only	None
	m.	.LTM	ST or MT	WT1+WT2	None
	n.	.TPU	ST	WT1+WT2	None
Group 5.	a.	.CMT	ST or MT	None	None
	b.	.HLT	ST or MT	None	None
	c.	.MTS	MT	None	None
	d.	.RST	ST or MT	None	WT
	e.	.CDC	ST or MT	None	None

INDEX OF TERMS

Term	PART	PAGE
Alternate unit (also Alternate tape unit)	V	18
APS (Assembly Programming System)	I	3
BASICPRG	II	5
BASICSUB	II	7
BIOS	III	14
CHKP (Check Point)	III	154
COBOL	I	3
Controlled execution	I	9
Debugging Format Mode	III	64
DEBUGGING Orders	III	57
Detail card	III	3
Detail Card Names (Used in Order)		
BLK .SGE	III	73
BRP .BPR	III	147
CDC .STG	III	90
CRT .DPG	III	66
DTR .DPG	III	66
EBL .SGE	III	73
LBC .PRL	III	45
.LTU	III	128
.MTG	III	137
LPI .LTM	III	150.5

Term		PART	PAGE
MCC	.STG	III	50
OPC	.MTG	III	139
OSR	.TPU	III	150.15
	.TCK	III	85.18
PIN	.TPU	III	150.12
	.TCK	III	85.18
PLG	.STU	III	101
PNG	.STG	III	91
PRG	.STG	III	89
PRD	.TPU	III	150.14
	.TCK	III	85.18
PRE	.MTG	III	140
PRI	.LTU	III	130
PRM	.LTU	III	131
PRR	.TPU	III	150.13
	.TCK	III	85.18
PRT	.LTU	III	130
RDC	.STG	III	88
SBL	.SGE	III	73
SIN	.STU	III	102
SLS	.SLS	III	112
SPC	.SPC	III	123
SRP	.STU	III	103
SSB	.PSI	III	85.10
STS	.STS	III	108
TLW	.TLW	III	30
TPC	.STG	III	89
TPM	.SGE	III	74
TRC	.DPG	III	66
UPC	.MTG	III	139
EIOS		I	4
End Card		III	3
File Table		V	15

Term	PART	PAGE
FORTRAN	I	3
ID Sequence code	III	3
(also Optional ID Code)		
ID Sequence Validity Switch	III	4
(also ID Switch)		
INITIALIZATION Orders	III	24
LBC card	V	17
LT (Library Tape)	II	8
Main unit		
(also Main tape unit)	V	18
Monitor	III	17
MT (Master Tape)	II	9
MT Control Area	III	17
MTM (Master Tape Mode)	III	16
Optional ID Code		
(also ID Sequence code)		
Order cards	III	2
Order Programs	III	22
Order Program Names		
.API Assembled Program Insert	III	41
.APP Assembled Program Punch	III	53
.APS Assemble APS Program	III	34
.BPR Basic Program Replace	III	146
.CDC Cards Duplicate	III	159
.CMT Comment	III	151
.COB Compile COBOL Program	III	36
.CRD Card Dump	III	82
.DPG Debugging Program Generate	III	63
.FOR Compile FORTRAN Program	III	37
.HLT Halt	III	152
.INT Initialize	III	24
.LPD Library Tape Print and Duplicate	III	150.1
.LTM Library Tape Merge	III	150.4
.LTU Library Tape Update	III	126
.MDP Memory Dump	III	84
.MTG Master Tape Generate	III	135

Term	PART	PAGE
.MTS Master Tape Stop	III	153
.PCC Punched Card Check	III	85.0
.PRL Program Load	III	45
.PSC Punched Subroutine Check	III	85.4
.PSI Punched Subroutine Insert	III	85.9
.RST Restart	III	154
.SBP Subroutine Punch	III	56.1
.SGE Samples Generate	III	70
.SLI Subroutine Library Insert	III	115
.SLS Subroutine Library Strip	III	111
.SPC Subroutine Peripheral Change	III	122
.SPD System Tape Print and Duplicate	III	94
.SPG Generate Sort/Merge Program	III	39
.STC System Tape Close	III	97
.STG System Tape Generate	III	86
.STS System Tape Strip	III	107
.STU System Tape Update	III	99
.TCK System Tape Checking	III	85.15
.TDC Tape Duplicate	III	57
.TDP Tape Dump	III	78
.TPU ST Program Update	III	150.8
.TLW Tape Label Write	III	29
Parameter cards	III	1
Pilot 0	III	7
Pilot 1	III	10
Pilot 2	III	12
PROGRAM TAPE GENERATE AND MAINTAIN Orders	III	86
Routing Code	III	18
Segment 00	III	135
Segment 99	III	136
Segment Load Routine	III	13
SPG (Sort Program Generator)	I	4
S/R (Subroutine)	I	4

Term	PART	PAGE
SRA (System Reserved Area)	I	10
ST (System Tape)	II	2
STG (Loader)	V	21
STM (System Tape Mode)	III	16
Supervisor	III	7
System Tape Generation	V	21
Trace Point	III	63
Trace Point Table	III	63
"TRANSLATE AND GO" Orders	III	34
Translator program	I	3
Type 1 programs	II	1
Type 2 programs	II	1
Type 3 programs	II	1
URA (User's Reserved Area)	I	10
USDA (User's Special Data Area)	V	8
User programs	I	7
Utility programs	I	4
Work Tape (WT)	I	7

DOCUMENT REVIEW SHEET

TITLE: ETOS Programmer's Manual
CODE: 3918106 B / Réf. 32.38.001 A

FROM:
Name :
Position:
Address:

Comments concerning this publication are solicited for use in improving future editions. Please provide any recommended additions, deletions, corrections, or other information you deem necessary for improving this manual. The following space is provided for your comments.

COMMENTS:

Fold on two lines shown on reverse side, staple, and mail.

Please cut along this line

FOLD

Postage

GENERAL ELECTRIC
INFORMATION SYSTEMS ITALIA

Ufficio Documentazione e distribuzione
Software del Marketing Support

via Pirelli, 32

20124 MILANO
(ITALIA)

FOLD

GENERAL ● ELECTRIC è marchio registrato della General Electric Company U.S.A.

GE 100 Line
DOS - Disc Operating System

Honeywell Bull

Code 3916163 F

Réf. 32.38.003 A

GE 100 Line
DOS - Disc Operating System

DECEMBER 1970

Honeywell Bull

Code 3916163 F

Réf. 32.38.003 A

PREFACE

This manual has been produced by Honeywell Information Systems Italia. The terminology used in this manual is according to the International Federation for Information Processing and the International Computation Centre (IFIP/ICC) Vocabulary of Information Processing.

Suggestions and comments pertaining to form, content, purpose, or use of this manual are invited. Comments may be sent on the Document Review Sheet which appears at the rear of this manual.

2nd Edition: supersedes 1st Edition dated May 1968.

© 1970 by Honeywell Information Systems Italia I[illegible].

INDEX

Page

PART ONE - GENERAL DESCRIPTION 1

CHAPTER 1. - INTRODUCTION 2

- 1.1. - General 2
- 1.2. - System Configuration 4
- 1.3. - Storage Requirements 4

CHAPTER 2. - DOS FUNCTIONS 5

- 2.1. - Program Compilation 5
- 2.2. - Program Testing and Debugging 6
- 2.3. - Program Storage 6
- 2.4. - Program Execution 7

CHAPTER 3. - SYSTEM DESCRIPTION 8

- 3.1. - Function of the Program Files 8
  - 3.1.1. - The System Disc (SD) 9
  - 3.1.2. - The Library Disc (LD) 10
  - 3.2.3. - The Master Disc (MD) 10
- 3.2. - Orders 12
- 3.3. - Execution of Orders by System Disc (or Library Disc) 17
- 3.4. - How the Master Disc Works 19

CHAPTER 4. - SYSTEM COMPONENTS 21

- 4.1. - Program File Structure 21

Page

4.1.1. - Program File Label 21
4.1.2. - Program File Index 22
4.1.3. - Program Structure 23
4.1.4. - The Subroutine Library 25

4.2. - Initialization Program 27

4.2.1. - Pilot Zero 27
4.2.2. - Bootstrap 28
4.2.3. - Basic Program 28

4.3. - Reserved Area 30

CHAPTER 5. - SYSTEM INITIALIZATION 36

5.1. - Conditions for initialization 36
5.2. - Initialization of System Disc and Library Disc 37

5.2.1. - Functions of Pilot 0 37
5.2.2. - Functions of Bootstrap 37
5.2.3. - Functions of Pilot 1 37
5.2.4. - Functions of Pilot 2 38

5.3. - Functions of segment load routine 40
5.4. - Basic Program HLT messages and instructions 41
5.5. - Pilot 0 Restart 43
5.6. - Master Disc initialization 44
5.7. - Monitor operation 45

CHAPTER 6. - GENERATION OF A SYSTEM DISC 50

6.1. - General 50
6.2. - The Load Order by Cards program 50

6.2.1. - Objectives 50
6.2.2. - System usage 51
6.2.3. - Parameter card 51
6.2.4. - Structure and execution 52
6.2.5. - Checks and error messages 53
6.2.6. - Example 53

Page

PART TWO - DOS ORDERS 57

•DNA - Disc Number Assignment 58
•INT - Initialize 61
•IND - Initialize Disc 65
•IDC - Initialize Disc by Cards 69
•FLW - File Label Write 74
•APS - Assemble APS Program 84
•COB - Compile COBOL Program 86
•FOR - Compile FORTRAN Program 88
•SGT - Generate Sort/Merge for Tape 90
•SPG - Generate Sort/Merge Program 92
•TAB - Translate TAB Program 94
•API - Assembled Program Insert 96
•PRL - Program Load 100
•APP - Assembled Program Punch 105
•PCC - Punched Card Check 108
•PTR - Program Transfer 113
•PCK - Program Checking 121
•GAF - Source Program File Generate 127
•LPC - Load Program by Cards 146
•DPG - Debugging Program Generate 155
•DIL - Disc Load 164
•DIP - Disc Print 171
•FLP - File Label Print 175
•CRD - Card Dump 178
•MDP - Memory Dump 181
•SPD - System Disc Print and Duplicate 183
•SDC - System Disc Close 188
•CZD - Cylinder Zero Duplicate 191
•SDU - System Disc Update 194
•SDS - System Disc Strip 204
•SLS - Subroutine Library Strip 208
•SLI - Subroutine Library Insert 212
•SPC - Subroutine Peripheral Compile 218

Page

●LDU - Library Disc Update 222
●MDG - Master Disc Generate 230
●MDS - Master Disc Stop 239
●DDP - Disc Print 240
●CMT - Comment 245
●HLT - Halt 246
●SDG - System Disc Generate 247
●RST - Restart 258
●SGE - Sample File Generate (on tape) 267
●TDP - Tape Dump 276
●TRP - Tape Record by Record Print 281
●TLW - Tape Label Write 285
●TDC - Tape Duplicate 293
●DFG - Diagnostic File Generate 302
●VLC - Volume Label Change 306

APPENDICES 309

APPENDIX A - LBC CARDS PROCESSING 310

A.1. - File opening 310
A.2. - LBC cards format 310
A.3. - Structure of file table, for tape files 313
A.4. - Structure of file table, for disc files 315
A.5. - LBC cards processing 317

APPENDIX B - PILOT ZERO CARD FORMAT 319

Page

DOS PROGRAMS : ALPHABETICAL ORDER

•API - Assembled Program Insert 96
•APP - Assembled Program Punch 105
•APS - Assemble APS Program 84
•CMT - Comment 245
•COB - Compile COBOL Program 86
•CRD - Card Dump 178
•CZD - Cylinder Zero Duplicate 191
•DDP - Disc Dump Program File 240
•DFG - Disc File Generate 302
•DIL - Disc Load 164
•DIP - Disc Print 171
•DNA - Disc Number Assign 38
•DPG - Debugging Program Generate 155
•FLP - File Label Print 175
•FLW - File Label Write 74
•FOR - Compile FORTRAN Program 88
•GAF - Source Program File Generate 127
•HLT - Halt 246
•IDC - Initialize Disc by Card 69
•IND - Initialize Disc 65
•INT - Initialize 61
•LDU - Library Disc Update 222
•LPC - Load Program by Card 146
•MDG - Master Disc Generate 230
•MDP - Memory Dump 181
•MDS - Master Tape Stop 239
•PCC - Punched Card Control 108
•PCK - Program Checking 121
•PRL - Program Load 100
•PTR - Program Transfer 113
•RST - Restart 258
•SDC - System Disc Close 188
•SDG - System Disc Generate 247
•SDS - System Disc Strip 204

			Pagina
•SDU	-	System Disc Update	194
•SGE	-	Samples Generate	267
•SGT	-	Generate SORT/MERGE for TAPE	90
•SLI	-	System Library Insert	212
•SLS	-	System Library Strip	208
•SPC	-	Subroutine Peripheral Change	218
•SPD	-	S. D. Print or Duplicate	183
•SPG	-	Generate SORT/MERGE Program	92
•TAB	-	Translate TAB Program	94
•TDC	-	Tape Duplicate	293
•TDP	-	Tape Dump	275
•TLW	-	Tape Label Write	285
•TRP	-	Tape Record by Record Print	281
•VLC	-	Volume Label Change	306

PART ONE. GENERAL DESCRIPTION

CHAPTER 1. INTRODUCTION

1.1. GENERAL

The Disc Operating System is a comprehensive software system made up of programs (pilots, compilers, parametered utility programs) and subroutines arranged according to certain criteria and usable in keeping with well-defined procedures.

The DOS is used for two specific activities, viz. program generation and program running.

The first of these activities, which consists of compiling, testing and debugging programs, takes place during system implementation and again whenever new services are programmed or existing services modified.

As this activity is of a one-time nature, it may be classified as "routine operation"

The second activity, however, consists of normal program running. As this activity is of a cyclic or repeated nature, it may be classified as a "non-routine operation".

The use of the DOS, whether for non-routine or routine work, achieves the following objectives:

A. Automatic linking between the various phases of an operation which reduces operator intervention to a minimum and consequently reduces the possibility of error. This means, for example, that it is possible to compile a program, generate sample files, start the compiled program and execute file or memory control printouts without involving the operator in the transfer from one phase to another.

It also means that during a non-routine operation an entire series of programs can be executed without interruption, by initially setting the linkage between a given program and the one following it.

B. Standardization of communications between the operator and the programs.
This means that whenever an error or alert occurs in any program, standard printouts indicate this status, or the program stops at fixed halt locations, which simplify the operator's duties to a good extent.

C. Programs independent of the principle peripheral units (magnetic tape units, card reader and printer) attached to the system.
To understand this point, it must be remembered that all programs using the DOS use discs (and possibly magnetic tapes), while for read and print operations they use routines stored at fixed addresses in the memory. Different versions of these routines are used according to the system configuration.
Consequently, a read or write instruction will always produce the same effect, regardless of which peripheral unit is used(x).
In short, this means that a given program can be executed on systems using different peripheral units.

(x) Regarding magnetic tapes, it is, of course, impossible to store data necessitating 9-track tape on a 7-track tape; the contrary is however, possible. As for discs, only one processing program is available as, up to the present time, only one type of disc unit is attached to the GE 115:

1.2. SYSTEM CONFIGURATION

The minimum system configuration required for DOS is :

- 8192 core positions, except for the orders •COB and •FOR which require 12288 core positions, and for the order •GAF which requires 16384 core positions.
- A card reader compatible with the system
- 2 magnetic disc units - DSU 130
- A Printer compatible with the system

1.3. STORAGE REQUIREMENTS

For systems using a Hollerith reader (Code H), the DOS requires 1600 core positions; for systems equipped with a BULL reader (Code T), the requirement is 1800 core positions.

This area will contain Input/Output management programs, i.e. a read routine, a write routine and a disc management routine. It will also contain read and write areas, segment load routine, a trascoding table for systems using a BULL reader (Code T). The contents of this area is described in section 4.3.

CHAPTER 2. DOS FUNCTIONS

2.1. PROGRAM COMPILATION

A primary problem solved by the use of the DOS is the compilation of program written in symbolic language.

The programmer will never write programs directly in machine language, but will use general languages such as TAB III and APS II or specialized languages such as SPG (Sort Program Generator).

Using the DOS, any program writtent in one of the specificied symbolic languages can be compiled.

A program written in APS requires a single pass to compile it in machine language. For a program writtent in TAB or SPG, two passes are required to compile the object program in machine language.

The TAB and SPG compilers generate an object program in APS language, which must, in turn, be translated into machine language by the APS compiler.

Passes from one phase to another are accomplished without operator intervention.

The DOS includes a comprehensive subroutine library. Using symbolic programs, subroutines are called implicitly in certain macro statements or explicitly through the SUB statement.

In answer to such calls, the compilers assemble the subroutine for the main program.

An object program can be made up of one or more segments.

The object program for a source program written in TAB or SPG language will always be made up of more than one segment.

The object program is produced by the APS compiler in a work area of the volume containing the program file (System Disc). Another operation inserts it in the program file so that it can be initiated.

2.2. PROGRAM TESTING AND DEBUGGING

DOS contains a comprehensive set of tools that give considerable aid during program testing and debugging.

Through the use of these tools, before actual program initiation the programmer can create sample disc or tape files. He can also request that his program be "traced", i.e. that all jump instructions executed be printed.

A compiled program can be regenerated in controlled form, i.e. so that it will accept correction cards in machine language and print the contents of the major memory areas at previously specified locations.

In addition, it is also possible to interrupt processing at any point during the test phase of a program for check-out purposes, such as file print-out and memory check.

2.3. PROGRAM STORAGE

After testing and debugging, a program should be re-writtent in symbolic language in its final form and then re-compiled.

The resulting program shoud be inserted in the appropriate file containing all the other programs that have been tested and debugged. This file (Library Disc) serves as a "collective file" for the EDP center. The contents of the LD are maintained by DOS.

2.4. PROGRAM EXECUTION

DOS is also a considerable help in normal management. In preparing a program for execution or in scheduling work for the day, the user must indicate which programs are to be executed. He establishes the order in which programs are to be run by setting the criteria which specify the linkages between programs and the point (s) for operator intervention, when required.

DOS combines these programs and makes up a special file, known as the Master Disc, which links each program with the next one by means of a Monitor which starts up the next program in accordance with the initial choice.

Linkage from one program to another occurs automatically without operator intervention. The operator intervenes only where specified by the user, e.g. to mount specific tapes or discs or to change the paper in the printer.

Once the Master Disc has started operating, it executes all the programs in the order specified by the user. In this way, once the user has established execution criteria for a given job and set up the Master Disc for this job, the program execution sequence can no longer be changed.

Each program can, however, be initiated individually by using the Library Disc.

CHAPTER 3. SYSTEM DESCRIPTION

3.1. FUNCTION OF THE PROGRAM FILES

In the previous section, reference was made to the function of the program file of the System Disc, Library Disc and Master Disc.

The file containing the programs is organized similarly to a data file. This file is identified by a file label, written on the first track of cylinder zero, which contains the name, date of creation and size within the volume. The file may occupy any volume area, but cannot be split up. The area occupied by the program file must consist of an integer number of cylinders.

At the beginning of each file there is an index of the programs (and subroutines, if any) contained in the file.

The program file is always one volume and is written in the sector mode. A volume cannot contain more than one program file. As a result, the System Disc, Library Disc, and Master Disc will always be stored on different volumes.

The same volume occupied by a program file can also be occupied by data files since the program file appears the same as any other file on the volume. The names System Disc, Library Disc and Master Disc are deceptive in that they give the impression that the entire disc is at the program's disposal. Although it is true, that a program file must always be accessible, both for routine and non-routine operations, this does not mean that the disc unit is completely available for programs.

3.1.1. The System Disc (SD)

The System Disc is the basic program file for DOS. It contains the complete software system and is used to derive the Master Disc or the Library Disc either directly or indirectly.

The System Disc contains the following:

a. Basic programs (i. e. system pilots, input/output management programs)

b. Subroutine library

c. Compilers

d. Parametered utility programs

e. Auxiliary and debugging programs

f. System initialization programs

g. System management programs

h. User's programs, if any.

Some of the more commonly used programs such as the Basic Program are functionally a part of the System Disc, but are written outside the program file's physical limits, viz. on tracks 1 through 9 of cylinder zero. Since the contents of cylinder zero is recorded on all the volumes during serializing, the fundamental programs of the DOS can automatically be used by the Library Disc and the Master Disc as well.

The System Disc is used for compiling, testing and debugging the user's programs.

The user programs are normally found on the System Disc only during the test phase. After the program has been completely checked and debugged, it is inserted on the Library Disc ready for future use.

A user program can, however, remain on the System Disc even though it is in its final form.

3.1.2. The Library Disc (LD)

The Library Disc is the file that contains the user programs that have already been debugged. It is always stored in a volume other than the volume in which the System Disc file is stored.

The contents of cylinder zero is identical in both volume, and the fundamental DOS program, such as the Basic Program, also form part of the Library Disc.

The Library Disc also contains all the programs that are required to set up the Master Disc and make it operational, which means that it also contains the specific Master Disc generation program and initialization program.

The Library Disc contains the following:

- Basic Program
- Fundamental DOS program required to set up the Master Disc
- All the user's programs in their final form.

Since the Library Disc contains the Basic Program, it can control the start and execution of the user's programs. It is not, however, possible for the Library Disc to achieve automatic linking to the execution of a series of programs; the Master Disc can do so.

3.1.3. The Master Disc (MD)

The Master Disc is a file containing all the programs linked together that constitute the routine job flow for a given period of time in the system: e. g. a day's work.

The programs forming part of the MD are taken either from the SD or the LD.
Two segments are added to each user program at the moment it is

inserted in the MD: one beginning segment containing the program start conditions; an end segment which starts the Monitor and determines its running conditions, i. e. the standards according to which it selects the next program to be started.

The Master Disc always contains the following programs:

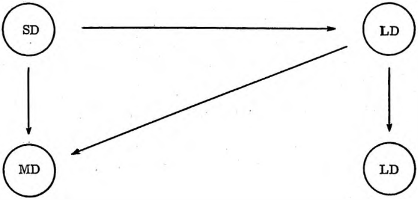
1. Basic program

2. System initialization programs

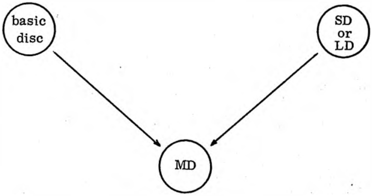
3. All the user programs that must be executed in that particular phase of the work.

In addition to the above programs, which form the essential contents of the Master Disc (MD), the user can insert the DOS programs he thinks necessary, e.g. the restart program (if restarts have been allowed for in the user programs), the file label writing program the disc print program, etc.

The flexibility of the system allows a Library Disc (LD) to be generated, or else a Master Disc (MD), with programs being transferred from one file to another, as shown below:



With three disc units available, the following procedure may be applied :



3. 2. ORDERS

To use the DOS, the user must indicate an appropriate sequence or order which is communicated to the system via a punched card known as an order card. For example, to compile a TAB program, a card containing the following indications in the first columns must be transmitted to the system:

●TAB

The DOS does not come to a halt after executing an order, but looks for another order card. It is therefore capable of executing a series of orders without coming to a halt.

The user must thus plan his work by preparing a deck of order cards.

For example, assuming that a program to be coded in APS is to be compiled and then test run,the following order cards must be communicated to DOS:

a. Program compilation

b. Insertion of the object program on the System Disc to allow execution

c. Generation of sample files, where required

d. Program execution.

Each of these orders is transmitted by one card.

Order cards are all marked with the character . (period) in the first column, followed by an alphabetic code of 3 characters which specifies the particular type of order.

In this case the orders would be coded as follows:

•APS - compiles an APS program

•API - inserts the compiled program on the SD

•DIL - writes the sample file on the disc

•PRL - starts the compiled program.

The information contained in the order card is not always sufficient for the DOS to carry out an order so that order requirements must be further specified in other cards.

In this way, the •DIL order must be followed by the cards containing the information to be stored; the •APS, and other compilation orders •TAB and •SPG, must be followed by the source program cards.

In this case, it may be said that order cards have to be followed by "detail cards" specifying the order requirements.

Still referring to the previous example, we note that the contents of the order •API is wholly specified in the order card itself and is not followed by any detail cards, while the order •PRL is followed by LBC opening cards only when it is desired to modify the program file table.

In short, three cases are possible:

a) The order card is never followed by detail cards

b) The order card is always followed by detail cards

c) The order card may be followed by detail cards.

In cases b and c, the end of the detail cards relative to a given order must be indicated by an order "End Card" to keep the next order card from being taken as a detail card and as such not being recognized.

In case c, when there are no detail cards, the order "End Card" must be present just the same, after the order card.

Referring again to the series of orders in the previous example:

•APS - compiles an APS program

•API - inserts the compiled program in the SD

•DIL - writes a sample file on disc

•PRL - starts the compiled program.

It is obvious that the order •API cannot be extended if the previous order has not been completed. In the same way, order • PRL cannot be executed if the orders •API and •DIL have not been executed correctly.

To achieve this linkage of orders, a specific order card code must be used: the "serial code". This code is used to identify a complete series of dependent orders, as explained above.

When there are consecutive orders marked by the same serial code, the DOS executes an order only if the previous order has been executed correctly.

If an order has no serial code, it is still executed. The presence of an order without serial code among other orders all having the same serial code does not interrupt the series itself.

To return to our example again.

We have four dependent orders, all of which are identified by the same serial code:

•APS - serial code AAA1

•API - serial code AAA1

•DIL - serial code AAA1

•PRL - serial code AAA1

These orders are connected to one another in such a way that failure to carry out one of them prevents the execution of those following.

Suppose we now wish to insert the Disc Print order (•DIP) after the order •DIL, in order to print the sample file.

We want this order executed even though the previous order was not completed. The order •DIP will have no serial code. The orders will then be arranged as follows:

•APS - serial code AAA1

•API - serial code AAA1

•DIL - serial code AAA1

•DIP - serial code b̸b̸b̸b̸

•PRL - serial code AAA1

The order •DIP is always executed, while the execution of the order •PRL remains conditioned by the correct execution of the order preceding the •DIP.

The series of orders is interrupted only when there appears an order with a different serial code.

Example:

●APS - serial code AAA1

●API - serial code AAA1

●DIL - serial code AAA1

●DIP - serial code b̸b̸b̸b̸

●PRL - serial code AAA1

●APS - serial code BBBB

●API - serial code BBBB

The second order ●APS is no longer connected to the preceding orders, but starts a new series.

Columns 73 to 76 are reserved for the serial code. Columns 77 to 80 must remain blank.

Columns 5 to 72 may be used to specify the order further.

The general layout of the order cards is therefore as follows:

Col. 1 - ● (period)

Col. 2 to 4 - order identification code

Col. 5 to 72 - other information (if any)

Col. 73 to 76 - serial code (if desired)

Col. 77 to 80 - b̸b̸b̸b̸

3.3. EXECUTION OF ORDERS BY SYSTEM DISC (OR LIBRARY DISC)

In the previous paragraph we saw how a series of orders is formulated.

We will now turn to the procedure used in executing the orders.

The SD, the basic program file of the DOS, can execute all possible orders. It is used to carry out what we have loosely termed routine operations, i.e. compilations, debugging of user's programs, building up and maintaining the Library Disc.

The SD therefore contains all the DOS programs. For each order (with the exception of •CMT, •HLT, and •MDP)there is a corresponding program on the SD.

Besides all these programs, the SD contains the Basic Program, which is the fundamental program for the whole system. In addition to fulfilling the functions of pilot program through reading and interpreting order cards and initiating the corresponding program to be executed, this program also contains the input/output control routines, i.e. those routines responsible for actual tape and disc management operations and the read and print routine.

The initialization of the system takes place with the start of the Basic Program. Its first operation is to read a card and check if it is an order card.
If not, it reads another card until it finds one that is recognizable as the order card, that is, marked with a character • in the first column. Then it prints the card. If the order is of the types •CMT, •HLT, or •MDP, it executes it directly; otherwise it retrieves the program from SD and starts it. At this point, the control of the system is transferred to the program, which reads and processes the detail cards, if the order requires it.

When it has been executed, each DOS program restarts the Basic Program, to which control of the system then returns. The program for the order •PRL is an exception to this, its duty is to start a user program. In this case, the user program will restart the Basic Program.

Let us now consider the case of orders connected to one another by means of a serial code. When the Basic Program reads an order card with a serial code for the first time, it saves the code in a service area that will be updated only when a card with different code is read. Every time the Basic Program encounters an order with a serial code, it compares this code with the one in the service area. If the two codes are equal , a specific condition will be tested to determine whether the order is feasible or not. This condition is called the "feasibility condition" and can assume two alternate values :

YES - the order is feasible (i. e. can be executed)

NO - the order is not feasible because a previous order of the same series has not been correctly executed.

The initial value of the feasibility condition is YES. It is set to NO by DOS programs when they encounter incidents or errors that block the execution or invalidate the results. Before restarting the Basic Program, the user programs can also force the feasibility condition to NO. The Basic Program, then, when ecountering an order with serial code equal to that of a previous order, tests the feasibility condition. If this is YES, it starts the order's program, if it is NO it goes on to seek another order card. When it reads a card with a serial code different from the one stored in the service area, it updates the service area and resets the feasibility condition to the values YES x.

---

x As will be explained later, in order to guarantee that the MD works correctly, the Basic Program sets (Basic) the feasibility condition to YES even when it starts the order •PRL.

A schematic diagram of SD operation is given in Figure 1.

What has been said for the SD is also valid for the LD, with this difference: in the LD, not all the DOS programs are present, but only those that allow the set up of the MD. However, all the DOS programs present on the LD can be started by the order cards and are processed by the LD in exactly the same way as the SD.

3.4. HOW THE MASTER DISC WORKS

The Master Disc is a program file specifically generated for a given type of work, for example, an entire service of an application system, or a day's work.
Contained on the Master Disc are all the user programs involved in the work to be executed and the more generally applied DOS programs.

The MD initialization takes place with the Basic Program start. The Basic Program reads and interprets the cards and starts execution of the orders. The MD, therefore, works in the same way as the SD until a user program is started through the order •PRL. At this point, the user's program runs until its function is completed, then the Basic Program transfers control to the Monitor. As a result of the options executed, when the program was inserted in the MD, the Monitor carries out one of the following functions:

1. Seeks and starts a program selected by the operator.
2. Seeks and starts one of two possible programs selected by the operator positioning Switch 2.
3. Reads an order card and starts executing the order's program or if the order is •PRL, starts the program indicated.
4. Halts due to end of processing.

5. Seeks and starts a program selected by the previous program, which stored the program name in the card read area.

In the third case, if the order is •PRL, the Monitor directly starts the program requested, otherwise control of the system returns to the Basic Program and remains there until a new •PRL order card has been read.
In cases 1, 2 and 5, before starting the next program the Monitor examines the feasibility condition. If this is NO, instead of starting the program requested, it restarts the Basic Program.

The feasibility condition may be set to NO by any user program.
In this case, none of the programs linked to it are executed. The feasibility condition is set to YES by the Basic Program each time the order •PRL is executed.

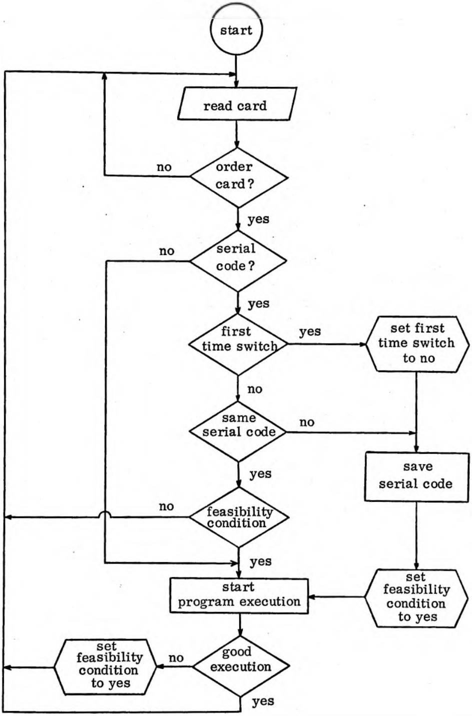

start
read card
order card?
no
yes
serial code?
no
yes
first time switch
yes
no
set first time switch to no
same serial code
no
yes
save serial code
feasibility condition
no
yes
start program execution
set feasibility condition to yes
good execution
no
yes
set feasibility condition to yes

Figure 1

CHAPTER 4. SYSTEM COMPONENTS

4.1. PROGRAM FILE STRUCTURE

Within DOS the program files on the LD, MD, and SD have a common structure with differences relative only to type of programs and number of programs present on each of them. In the following description of the program file, these differences existing between the three discs are pointed out.

4.1.1. Program File Label

The Program File is identified by a label in the file table, written in 6 bit mode in one of the sectors from 2 to 17 of cylinder zero, and having the following structure:

CHARACTER (S)	DESCRIPTION
1 - 4	HDR1
5 - 21	File identification the first two characters must be LD, MD, or SD (for Library Disc, Master Disc, or System Disc respectively).
22 - 41	Not used
42 - 47	ƀYYDDD - creation date
48 - 53	Not used
54	P (= Program file) file type code

CHARACTER(S)	DESCRIPTION
55 - 57	nnn - first cylinder address[*] reserved for the file
58 - 60	nnn - last cylinder address reserved for the file
61 - 65	Address of the first free sector in the file area
66 - 70	Address of the last page of the subroutine library. If this page is missing, address of the first sector after the longest sector set aside for a subroutine index, if one exists (the indication b̸ is only applicable to the SD)
71 - 100	Not used.

4.1.2. Program File Index

The first part of the disc area reserved for the program file is occupied by the program file index. This index is written in 6 bit mode, three sectors in a page.
Two tracks are reserved for the index on the LD and SD and a single track on the MD.
The end of the index is indicated by the character E. As 15 characters are allowed per program, the LD and SD can contain a maximum of 259 programs and the MD a maximum of 119. The index has the following structure:

---

[*] The cylinder address is the address of the first sector of the cylinder. In this case, only the first three characters are expressed, as the last two characters are always zero.

CHARACTER(S)	DESCRIPTION
1	Program type code; this code can have one of the following two values: 1 - DOS program 3 - user program
2 - 9	Program name, in alphanumeric characters
10	Program code: Code C if the program is generated in controlled form (see order •DPG). If the program is in normal form, this code indicates the use of basic programs on tape and on disc. b̸ = program use tape management routines only. 1 = program uses disc management routines only 2 = program uses both routines 3 = program uses neither of the both routine.
11 - 15	Disc address of the first page of program, which contains the program label.

Programs appear in the index in the same order in which they were written on the file.

4.1.3. Program Structure

On the LD and MD, programs immediately follow the program file index, while on the SD, they follow the subroutine library. Program are written in 8 bits mode in pages of variable length.

Each page of a program is preceded by a page label containing pertinent information about the following page. The first of these page labels also contains information pertinent to the entire program and is, therefore, called the program label.

The address of program label is contained in the file index with the program name.

Each page label occupies a sector and has the following format:

CHARACTER(S)	DESCRIPTION
1	Segment indicator in binary: 00 not last page of segment, FF last page of segment
2 - 3	Segment number in decimal
4 - 5	Length of page minus 1 in binary
6 - 7	Memory address of page in binary
8 - 12	Sector number of label for next page 00000 No more pages.
18 - 75	Not used by page label

The program label also contains the following information:

CHARACTER(S)	DESCRIPTION
51 - 52	Address of the beginning of the file definition table in binary
53 - 54	Program start address in binary
55 - 56	Last memory address + 1 of program in binary
57 - 62	0YYDDD - Insertion date

4.1.4. The Subroutine Library

The subroutine library is only found on the SD and immediately follows the Program File index. It is written in 6 bit mode and preceded by an index. There are ten sectors in a page. Each record has the following structure:

CHARACTERS	DESCRIPTION
1	*
2 - 6	Sector number of first page of subroutine
7 - 12	Subroutines name
13 - 16	Subroutine length in unpacked binary

A maximum of 6 subroutines can be indicated. Information regarding these possible subroutines is given below. They must be in the following format:

- 6 characters for the subroutine name
- 4 characters for the subroutine length.

Index records therefore vary in length from 16 to 64 characters. The end of meaningful information on a page is marked by an asterisk, and the end of the complete index by two asterisk. Five tracks are set aside for the index.

Subroutines immediately follow the subroutine index and cover a whole number of variable length pages with a maximum length of 12 sectors. The first characters of each page contain the following information:

CHARACTERS	DESCRIPTION
1 - 5	Disc address of the next page
6 - 11	Subroutine name
12	Number of subroutines required by this subroutine (from 0 to 6)
13 - 18	⌀YYDDD - subroutine insertion date.

Next follows a list of any subroutines called. The list is in the following format:

- 6 characters for the subroutine name
- 1 character for the sequence number

After this list for which 42 characters are always reserved, comes the following information:

- 4 characters - Length of the part of the subroutine inserted in the page
- 4 characters - Memory address of this page.

Then follows the actual text of the subroutine. Each subroutine has been translated into basic language and compiled relative to a starting address of zero. The result is that these addresses are all relative and must be adjusted, when assembling the subroutine to the main program, by adding to them the actual memory address.

Subroutines are written in 6 bit mode and are therefore unpacked. The comment area assumes one of the following values to accomplish this adjustment:

a. 4 - no adjustment

b. 5 - adjustment required during assembly phase

c. A - two characters preceding call subroutine sequence number. This value is written with the no adjustment value 4.

d. B - peripheral unit number which varies according to the system configuration.

The end of meaningful information on a page is marked by the character R and the end of the complete subroutine by the character E.

The subroutine library occupies a whole number of tracks. The sector number of the last subroutine track is contained in the program file label.

The following table summarizes the make up of program files.

Type of File	Contents	Number of Tracks	Mode
SD	1. Program File Index	2	6 bit
	2. Subroutine File Index	5	6 bit
	3. Subroutine Library	n	6 bit
	4. Programs	n	8 bit
LD	1. Program File Index	2	6 bit
	2. Programs	n	8 bit
MD	1. Program File Index	1	6 bit
	2. Programs	n	8 bit

4.2. INITIALIZATION PROGRAMS

4.2.1. Pilot Zero

Pilot Zero is a one card hexadecimal program which initializes the system.

This program varies according to the actual unit on which the program to be initialized is mounted (from then on this file will be called the system's basic program file). It is started manually by pressing the RESET, LOAD 2 and RUN buttons, in that order. It reads the Bootstrap in sector 00020 of the disc which contains the program file and starts the operation.

Pilot Zero is permanently stored in the memory where it occupies the first 40 positions.

4.2.2. Bootstrap

The Bootstrap program is the link between the Pilot Zero and the Basic Program. Along with the Basic Program, the Bootstrap is stored on cylinder 0 of each volume and takes up disc sector 0020.

The Bootstrap reads the following sector containing the list of programs present on cylinder 0, then starts up the basic program.

The Bootstrap is, then, overlapped.

4.2.3. Basic Program

The Basic Program consists of the following segments:

1st segment : Pilot One

This is the first of the two Basic Program Pilots. It is started by the Bootstrap.

Pilot One reads all order cards, checks their validity and checks feasibility condition, directly executes the orders •CMT and •HLT, starts the memory dump routine for an order •MDP and the programs associated with the orders •DNA and •LPC (which are also stored on cylinder 0), and loads Pilot 2 in all other cases.

After execution, Pilot 1 is overlapped.

2nd segment : Data and Service Area

The data and service area segment contains all the data relevant to the particular system configuration, the DOS service area, the card read and print areas and the transcoding program for systems equipped with a BULL card reader.

This segment is loaded by Pilot 1 and always remains in memory from decimal addresses 100 to 511.

3rd Segment : Read and Print Routine

This routine is used by the DOS for card read and printing operations. This routine varies with the particular type of card reader and printer attached to the system.

The read and print routine is used by all DOS programs and is called by user programs through macro-instruction *LBSS. (see the manual "APS II Basic Language").

The read and print routine always remains in memory during the execution of DOS programs and user programs.

4th Segment : Memory Dump Routine

This routine is started by Pilot 1 when an order •MDP is read.

5th Segment : Magnetic Tape Routine

The tape routine executes all the physical operations related to magnetic tape processing: read, write, rewind. Logical file processing subroutines, i.e. subroutines for macro-instructions GET, PUT, OPEN, etc., use this routine. (See the manual "APS II Basic Language").

The tape routine is loaded into memory above the read and write routine and may be overlayed by programs which do not process magnetic tape files.

6th Segment : Segment Load Routine and Disc Routine

The segment load routine is used by DOS as well as user programs to seek and start a program segment.

The disc routine carries out all the physical operations related to the handling of magnetic disc units: read, write, position, or control. The logical file processing subroutines and DOS programs use this routine.

This segment always remains in memory.

7th Segment : Pilot 2

Pilot two is loaded and started by Pilot One for certain orders after they have been checked for feasibility condition. It seeks and starts the associated program to be executed.

In addition to the Bootstrap and the Basic Program, cylinder 0 contains the programs associated with orders •DNA (which assigns logical numbers to disc units) and •LPC (which loads programs from cards).

Although these programs are not part of the Basic Program, they are located in the memory between the 5th and 6th segments.

4.3. RESERVED AREA

DOS system requires a reserved area of 1800 characters, for a system configuration using a BULL reader.

The size of this area is increased by 500 characters for mixed magnetic tape-disc programs.

The contents of this area is specified in the following table:

ADDRESSES		LL	CONTENT
BEGIN	END		
0000	0027	40	Pilot Zero.
0028	002D	6	Day's date, placed by the order •INT.
002E	0031	4	Decimal counter of the restart points. It is incremented by 1 by all the user programs when they execute a macro-statement CHKP or DCKP.

ADDRESSES		LL	CONTENT
BEGIN	END		
0032	0033	2	Control area. The first position is used by Pilot 1 of the Master Disc to check if the order card has already been read by the Monitor. The second posizion is used for controlling the feasibility condition.
0034	0053	32	Area reserved for the user loaded by the orders •INT and •PRL or from the segment 00 of a type 3 program.
0054	0057	4	Storage area for serial code.
0058	005F	8	Name of program being executed.
0060	0061	2	File table address.
0062	0063	2	First address left free by the Basic Program.
0064	0066	3	Free positions.
0067	0067	1	Hexadecimal number of punch, if provision made for it in the system configuration.
0068	006B	4	Jump Instruction for starting the mag. tape subroutine.
006C	006F	4	Jump Instruction for starting the read and print subroutine.
0070	0073	4	Jump instruction for starting the segment load routine.

ADDRESSES		LL	CONTENT
BEGIN	END		
0074	0077	4	Jump instruction for restarting tape routine after normal call with no indication of re-entry or parameters. This instruction can be used for continuous reading.
0078	007B	4	Jump instruction for starting disc routine.
007C	007D	2	Address of first memory position that can be used by the programs, i. e. 1600 for systems using a Hollerith reader, 1800 for systems using a BULL reader. This address does not include any tape routine.
007E	007E	1	Store dimension in multiples of 256. This data is exepressed in binary form.
007F	007F	1	The letter H, for System using a Hollerith reader, the letter B, for Systems using a BULL reader.
0080	0080	1	Logical number of the disc unit containing the initialized program file.
0081	0083	3	Free positions.
0084	0084	1	Number of tape units connected to system.
0085	0085	1	Free positions.
0086	00D5	80	Card read area.
00D6	00FB	38	Area at diposal of subroutines.
00FC	00FF	4	Re-entry jump after jump and return (instruction JRT.)

ADDRESSES		LL	CONTENT
BEGIN	END		
0100	016D	110	Area reserved for read and print routine, for a system using a Hollerith reader, and for the transcoding table for a system using a BULL reader.
016E	0177	10	Correspondence table between the logical and physical numbers of the disc units.
0178	0181	10	Extension of 0100 to 016D.
0182	018F	14	Transcode routine for systems using a BULL reader. For systems using Hollerith readers, the transcode routine is substituted by a re-entry jump.
0190	01BF	48	Extension of 0100 to 016D.
01C0	01C7	8	Correspondence table between logical and physical numbers of the tape unit.
01C8	01CD	6	Tape routine HLT instruction, signalling persistent reading or writing error after five efforts; followed by a jump instruction to restart attempts.
01CE	01D3	6	HLT blocking instruction belonging to tape routines signalling unit out of services, followed by a jump instruction that ends back to the same HLT.
01D4	01D9	6	Tape routine HLT instruction, signals error in block number after five attempts to find the exact block. It is followed by a jump instruction to restart attempts.

ADDRESSES		LL	CONTENT
BEGIN	END		
01DA	01DF	6	HLT instruction signalling disc unit out of service. After the first attempt, DOS programs are skipped. It is followed by a re-entry to the program.
01E0	01E5	6	Disc routine HLT instruction signalling persistent error, after five attempts. It is followed by a jump instruction to restart attempts.
01E6	01EB	6	Disc routine instruction HLT signalling unit out of service. It is followed by a jump instruction restarting the routine for a new attempt.
01EC	01FF	20	Area reserved for HLT instructions of the file logical handling subroutines (see "The APS II Language").
0200	0201	2	Area at disposal of printer controller.
0202	02A1	160	Print-out area.
02A2	02A5	4	Instruction setting the feasibility condition to NO. The DOS and user's programs jumpt to this address in the case of error.
02A6	02A7	2	HLT instruction signalling the previous operation.
02A8	02AF	8	Jump instruction restarting Pilot Zero. If the system control is the MD, at this address there is a jump to the segment load routine, followed by the parameters for starting the segment (segment number and starting address).

ADDRESSES		LL	CONTENT
BEGIN	END		
02B0	02B4	5	Starting address of the check point file.
02B5	02B9	5	Program label address of the program being executed.
02BA	02BB	2	Number of segment in execution.
02BC	02BF	4	Instruction setting the feasibility condition to NO. It is used by the segment load routine if the segment cannot be found.
02C0	02C1	2	HLT instruction signalling previous operation.
02C2	02C5	4	Jump instruction re-starting Pilot Zero.
02C6	02CF	10	Instruction setting the feasibility condition to NO. It is used by the segment feed routine in cases of persistent error in disc operations. Like the previous one, it is followed by a HLT and a re-entry jump to Pilot Zero.
02D0	0833 (08FB)*	1380 1580*	Segment feed routine, disc routine, tape read and print routine.

N. B. : (*) Starred data apply to systems using a BULL reader.

CHAPTER 5. SYSTEM INITIALIZATION

5.1. CONDITIONS FOR INITIALIZATION

Initialization of DOS depends on two preliminary conditions: preparation of a deck of order cards and detail cards for the work to be carried out, and preparation of the program file which will execute the work.

The type of program file to be prepared depends on the nature of the work to be carried out : for compilations, tests and debugging, the SD is used; for routine work, the MD is used.

Certain types of processing require two program files, as when a program is transferred from the SD to the LD using the order •LDU but in any case a single file is initialized and serves to control the entire processing operation: this file is called the basic system file and the volume on which it is written is called the basic disc.

If an SD or LD file is initialized, work sequence is entirely dependent on order cards. If the MD is initialized, however, processing is carried out through appropriate inter-program linkage. In this case, as well, processing will still be controlled by an order card.

Before any processing can take place, the orders •DNA and •INT must be executed.

Pilot Zero is located at the head of the card order deck. It reads on the unit housing the file to be initialized.

5.2. SYSTEM DISC AND LIBRARY DISC INITIALIZATION

5.2.1. Functions of Pilot 0

Pilot Zero is loaded into memory and started by means of the manual initialization operations (RESET, LOAD 2, and RUN). Pilot Zero reads the Bootstrap from sector 00020 of cylinder zero. The Bootstrap will then start Pilot One of the Basic Program.

Pilot Zero is always present in memory's first forty locations and is restarted each time a new order is to be executed. While a program is running, the processing can be interrupted and the next order can be started by pushing the buttons RESET and RUN, unless the first forty memory locations have been altered.

5.2.2. Functions of Bootstrap

The Bootstrap reads the directory of programs written on cylinder zero, retrieves the first segment of the Basic Program (Pilot One) and starts it.
A persistent read error halts at location 012C. Pilot Zero is restarted by pressing the RUN button.

5.2.3. Functions of Pilot 1

Pilot One carries out the following operations:

a) Introduces to the memory the second and third segments of the Basic Program (data and service area and read print routine).

b) If the MD file is initialized, checks that the order card has not yet been read by the program Monitor (value 01 at memory position 0032), otherwise it moves right on to the next operation.

c) Reads cards into the read area until an order card can be identified (character • in the first column) and prints the card preceding it with three asterisks and the day's date.

d) Checks the order feasibility condition via memory position 0033. This position may contain two different values:

00 = Feasible
01 = Not feasible

If the order cannot be executed, it prints the following message:

5010* ORDER NOT EXECUTED

and seeks another order card.

See section 3.3. for feasibility processing.

e) It examines the contents of the order card.

If the order is •CMT, the comment has already been printed and the program starts looking for another order card. If it is an •HLT card, it stops at address 01D6. Pressing RUN causes the program to seek a new order card. If the order is •MDP, it reads and starts the memory dump routine (4th segment). If the order is •LPC or •DNA, it loads the associated program together with segments 5 and 6 and starts it. For all other orders, it loads the tape routine, the segment load routine, the disc routine (6th segment), it then loads and starts Pilot Two (7th segment). At this point Control is transferred to Pilot Two.

When an error or out of order condition occurs during a disc operation, Pilot One halts at address 01CA. Pressing RUN causes the operation to be tried again.

5.2.4. Functions of Pilot 2

Pilot Two carries out the following operations:

a) It checks the logical numbers of the tape units connected to the system, if any.

This check is not carried out if a GUN 7/9 tape unit controller is being used.

b) Load the logical disc unit number showing the initialized file at memory location 0080 and in the segment feed routine parameters. The physical number of the unit is obtained from Pilot Zero.
If the logical number is not included in the correspondence table (at address 016E), it prints the following message:

5113 * MAIN DISC NUMBER ERROR

and halts at location 0000.

c) Searches segments 2 to 17 of cylinder 0 for the program file label, i. e. the label with SD, LD or MD as its first characters.
If the program label is not found, it prints the following message:

5114 * PFL NOT FOUND

and halts at location 0000.

d) if the program file label is found, it searches the program file index for the program associated with the order. If the program does not exist, Pilot Two prints:

5111 * ORDER UNKNOWN

This sets the feasibility condition to "NO" and halts at location 02A6. Pressing RUN restarts Pilot Zero. If, during the search, Pilot Two does not find the end of index character E, it prints:

5112 * END OF PROGRAMS TABLE NOT FOUND

This sets the feasibility condition to NO and halts at location 02A6. Pressing RUN restarts Pilot Zero.

e) If it has found the name of the program to be started in the index, Pilot Two transfers the program label address, located in the index, to the service area reserved for the segment load routine. It then sets all the memory that can be used by the programs to value 0A and starts the segment load routine.

If a persistent error in disc operation occurs, Pilot Two prints:

```
5101 *  D. UNIT N PERSISTENT ERROR
```

This sets the feasibility condition to "NO" and halts at address 02A6. Pressing RUN restarts Pilot Zero.
If the disc unit signals out of order, Pilot Two prints:

```
5103 *  D. UNIT N OUT OF ORDER
```

and halts at location 01DA, to allow the operator to put the unit back in service.

If the unit is still out of order when an attempt is made to retry the operation, the feasibility condition is set to "NO" and the program halts at 02A6.

5.3. FUNCTIONS OF SEGMENT LOAD ROUTINE

The Segment Load Routine searches for a program segment, loads it into memory, and starts it. The Segment Load Routine compares the segment number required with the last segment number read. If the comparison gives a less than or equal to result, the search starts from the beginning of the program; otherwise the search continues from the last segment started. For example, assume a program of 5 segments (numbered 1, 2, 3, 4 and 5) and segment 3 was the last segment read.

If the required segment is not found, the feasibility code is set to "NO" and the routine halts at location 0200. An error in disc operations also sets the feasibility code to "NO", but halts at location 02CA. Pressing RUN restarts Pilot Zero for both cases.

5.4. BASIC PROGRAM HLT MESSAGES AND INSTRUCTIONS

Error messages :

5101	x DISC UNIT N PERSISTENT ERROR	Persistent error on disc unit N. Processing halts at address 02A6. Pressing RUN starts search for a new order card.
5103	x DISC UNIT N OUT OF ORDER	Disc unit N is out of order. The program stops at address 01DA until the operator puts the unit back in order. If the same message appears again when processing is restarted the program halts at address 02A6. Pressing RUN starts search for a new order card.
5110	x ORDER NOT EXECUTED	The last order read cannot be executed. (Feasibility condition:NO). The program goes on to the next order card.
5111	x ORDER UNKNOWN	The requested program is absent from the program file. Processing halts at address 02A6. Pressing RUN starts search for a new order card.
5112	x END OF PROGRAM TABLE NOT FOUND	The end character E is not present in the file index. Processing halts at address 02A6. Pressing RUN starts search for a new order card.

5113	x MAIN DISC NUMBER ERROR	The logical number of the unit initialized cannot be found. The program halts at address 0000. Restart Pilot 0 and repeat •DNA order.
5114	x PFL NOT FOUND	No program file is present in the initialized volume. Processing Halts at address 0000.

HLT Instructions:

HLT ADDRESS	REASON	OPERATOR ACTION
0000	Blocking error, indicated by a printed message.	
012C	Read error during Bootstrap execution.	Pilot 0 is restarted by pressing RUN.
01D6	Pilot One stopped by a HLT order.	Press RUN. Pilot One will search for a new order card.
01CA	Read error or disc unit out of order in Pilot One.	Press RUN to retry operation.
01DA	Disc unit out of order in Pilot Two.	Correct the condition and press RUN.
02A6	A printed message indicates the reason for the halt.	

HLT Address	REASON	OPERATOR ACTION
02C0	Required segment not found.	Press RUN to restart Pilot Zero. The feasibility condition is "NO".
02CA	Disc unit error in segment load routine.	Press RUN to restart Pilot Zero. The feasibility condition is "NO".
01CC	Two or more tape units with the same physical number.	Press RUN to retry test. To continue without a test, set SWITCH 2 and press RUN. Another HLT takes place at address 01CC so that the operator can set SWITCH 2 to its original position.

5.5. PILOT 0 RESTART

When they have been executed, all the programs jump at address 02A8 (x) (decimal, 680) where the instruction that restarts Pilot 2 is located.

If the system is MD-controlled, however, the instruction used to start segment 99 is at this address. Segment 99 will in turn start the Monitor.

(x) This jump is made in the user programs by means of the macro-statement STOP (see APS II Language manual).

During the execution of a DOS program, if errors are encountered which block the execution or invalidate the results, the program jumps at address 02A2 (674 decimal); the instruction setting the feasibility condition to NO is at this address. The program then halts at address 02A6. Pilot 0 is restarted by pressing RUN.

In case of error a user program may also jump at address 02A2 instead of 02A8. This jump may be used to block the execution of MD programs which are linked to the user program.

5.6. MASTER DISC INITIALIZATION

The Master Disc is initialized by starting Pilot 0. As has already been seen in the case of the System Disc, Pilot 0 stores and starts the Bootstrap program.

The only difference compared with initializing the System Disc is that Pilot 1 checks the content of memory location 0032 before reading an order card. If this content is 01, it means that the card has already been read by the Monitor of the previous program. In such case, Pilot 1 resets the content of this location to 00 and then goes on to check the order card which is already in the read area.

Except for this difference, the MD runs in exactly the same way as the SD until the •PRL card is read. This card starts the first user program.

The type-1 program pertaining to the •PRL order is different from the one present in the System Disc. Before starting the required program, it modifies the memory locations 02A8 - 02AF by substituting a jump to call segment 99 in place of the Pilot 0 restart jump.

5.7. MONITOR OPERATION

The Monitor may carry out various functions according to the options specified when the MD is generated. Its running conditions depend on the preparation code written in segment 99 of each program. These running conditions may be one of the following:

1. Start a type 3 program selected at the time of MD generation.
2. Start a type-3 program selected according to how Switch 2 is positioned; the choice is made between two programs indicated while the MD is being generated.
3. Read an order card. If this is a •PRL, the Monitor starts the program indicated; otherwise it starts Pilot 0 after setting to 01 the value of memory location 0032.
4. Execute the order •MDS (Master Disc Stop) at end of processing.
5. Start a type-3 program selected by the same program to which segment 99 belongs; the name of the program is stored in the first location of the card read area.

The Monitor first checks the feasibility condition. If it is NO, it starts Pilot 0; then the linking provided by the preparation codes 1, 2 and 5 does not take place.

If the preparation code is 4, the order •MDS is still executed.

If the feasibility condition is YES, the Monitor prints the following message:

```
7599 x END OF PROGRAM X
```

where X is the name of the program that has just been executed. Then the Monitor follows different ways depending on the preparation code:

A. Preparation code 1

It prints the following message indicating the program to be started.

```
7521  x NEXT PROGRAM IS X
```

It then seeks and starts the program mentioned.

B. Preparation code 2

It prints the following message, on the basis of which the operator will position Switch 2

```
7522  x NEXT PROGRAM IS X WITH SWITCH 2 OFF/Y WITH
        SWITCH 2 ON
```

The Monitor then comes to a HLT at address 01D8 and waits for operator action. The operator must position Switch 2 as required : i. e. he must turn on the switch if program Y is to be executed and turn it off if program X is to be executed.

After pressing RUN, causes seeking and starting of the program selected.

C. Preparation code 3

The Monitor first prints out the following message:

```
7523  x NEXT PROGRAM SELECTED BY ORDER CARD
```

and goes on to seek an order card.

If the order cards •PRL it is printed and its serial code, if any, is moved to the appropriate storage area.

The Monitor then checks the validity of the •PRL card and of any LBC cards, which may be present.

If any errors are found, it prints one of the messages listed in the next pages and then jumps at address 02A2 (feasibility condition: NO) after resuming the Pilot 0 restart jump at address 02A8.

If no error is found it seeks the program mentioned in the •PRL card; if there are LBC cards the program file table, if any, is updated. The Monitor then starts segment 01 of the program proper. The execution of segment 00 is not needed since the start conditions are already contained in the •PRL card.

D. Preparation code 5

The Monitor prints the following message, which mentions the program to be started.

```
7525 x NEXT PROGRAM   X SELECTED BY ENDED PROGRAM
```

The name of this program has been placed in the read area by the program that has just been executed. The Monitor then seeks and starts the program to be executed.

The following messages may appear during Monitor execution:

```
7501  x UNIT X PERSISTENT ERROR
```

Persistent error in unit X. The Monitor halts at address 02A6.
Feasibility condition: NO. Pilot 0 is restarted by pressing RUN.

```
7502  x UNIT X OUT OF ORDER
```

Unit X is out of order. The Monitor halts at address 02A6.
Feasibility condition: NO. Pilot 0 is restarted by pressing RUN.

```
7510  x PROGRAM NNNNNN NOT FOUND
```

Indicates that the program to be started has not been found.
The Monitor halts at address 02AC.
Feasibility condition: NO. Pilot 0 is restarted by pressing RUN.

7511 x SUB-ORDER CARD ERROR

The identification code of a detail card is incorrect. The Monitor halts at address 02A6.
Feasibility condition: NO.
Pilot 0 is restarted by pressing RUN.

7512 x SUB-ORDER CARD COLUMN NN ERROR

The field starting from col. NN of a detail card is incorrect.

7513 x YYDDD/START OF PROGRAM X

Indicates start of a program. The day's date is printed.

7515 x ORDER CARD COLUMN NN ERROR

Error in column NN of card •PRL.

7516 x CHECK-POINT FILE TOO SMALL

The size of volume zone reserved for the CHECKPOINT file is not sufficient. The Monitor halts at address 02A6. Feasibility condition: NO.
Pilot 0 is restarted by pressing RUN.

7521 x NEXT PROGRAM IS X

Indicates the next program to be executed.

7522 x NEXT PROGRAM IS X WITH SWITCH 2 OFF/Y WITH SWITCH 2 ON

Operator action is requested for selecting the next program to be executed.
If program X, press RUN;
if program Y, turn on Switch 2, then press RUN.

7523 x NEXT PROGRAM SELECTED BY ORDER CARD

Indicates that the next program to be executed will be specified by an order card.

7525 x NEXT PROGRAM X SELECTED BY ENDED PROGRAM

Indicates the next program to be executed.
This program is selected by the program that has just been executed.

7599 x END OF PROGRAM X

Indicates that program X has been executed.

While order •MDS is being executed, the following message may appear:

7601 x MASTER DISC STOP

Processing halts at address 01D6.
Pilot 0 is restarted by pressing RUN.

Any HLT requested before a program start takes place at address 01D2.

CHAPTER 6. SYSTEM DISC GENERATE

6.1. GENERAL

To generate a System Disc means to personalize or adapt a standard System Disc to the desired configuration. Such generation is done by the program •SDG written on the standard System Disc.

Two methods are provided for calling and initiating the System Disc Generate program:

a) The order card •SDG

b) The "Load Order by Cards" program punched on cards.

Method a) is used when the configuration included in the Standard System Disc is the same as the one actually used by the system for generation; otherwise, the program Load Order by Cards is used.

Method a) is explained in the second part of this manual by the DOS control orders;
method b) is described in the following paragraph.

6.2. "LOAD ORDER BY CARDS" PROGRAM

6.2.1. Function

This program, punched on cards, core loads and initiates the •SDG program to personalize the System Disc on which the •SDG program itself is written. It is made use of a Standard SD containing a Basic Program which may correspond to a configuration other than the one actually used by the system.

6.2.2. System Unit Required

The following peripherals are required:

- a disc unit containing the standard System Disc to be personalized
- a card reader
- a printer.

6.2.3. Parameter Card

The parameter card is read-in as a patch card at the end of the Load Order by Cards program and precedes the end of program card (FF punched in columns 70 and 80). It contains the parameters required by the program itself and defines the system configuration.

It has the following format:

Col. 1 - 4	program code 0396
Col. 5 - 6	patch length in hexadecimal: 06
Col. 7 - 10	patch first character address in hexadecimal: 0200
Col. 11 - 12	reader type: b̸1 = CRZ100 CRZ120 b̸2 = to be assigned b̸3 = to be assigned b̸4 = CRP100
Col. 13 - 14	reader connection number (in hexadecimal) or blank when the reader has an integrated controller (that is b̸1 in columns 11 and 12)
Col. 15 - 16	printer type: b̸1 = PRINTER 300, PRINTER 600 b̸2 = PRINTER 1100 b̸3 = MB (PRT105)

ƀ4 = to be assigned
ƀ5 = to be assigned

Col. 17 - 18	printer connection number (in hexadecimal) or blank when the printer has an integrated controller (that is ƀ1 in columns 15 and 16.
Col. 19 - 20	punch code: blank = Hollerith code any punch = Bull code
Col. 21 - 22	disc unit connection number (in hexadecimal)
Col. 23 - 80	not used.

6.2.4. Description and execution

The Load Order by Cards program located at address 1800 (decimal) is a segment which includes the segment-load routine and the disc subroutine in addition to the program proper.

The program generates in the store some of basic program segments adding to the segment-load routine and the disc routine the read and print routines retrieved from the System Disc •SDG routine and adapts them to the configuration according to the parameter card; the program also sets up control to the tape and disc management subroutines and HLT statements to signal disc errors. Then it loads and initiates the •SDG routine to personalize the System Disc.

Using the Load Order by Cards program requires the following cards:

- Loader
- Load Order by Cards program with parameter card inserted immediately before the end of program card FF
- Detail cards for the •SDG control order.

6.2.5. Checks and error messages

On the basis of checks performed on the disc unit the program may halt at address 01E0 or 01E6.

When an error has been detected in one of the two fields of the order card which concern the type of reader or printer the program halts at address 0206.

If the ●SDG program is not recorded on the System Disc, the program halts at address 020C.

At end, the program halts at address 0212 and prints out the following message:

```
4099 * END OF LOAD ORDER BY CARDS
```

6.2.6. Example

A System Disc is to be adapted to a configuration made up of an 8,192 location memory connected to a CRP100 (connection number 40 in hexadecimal) used as card reader/punch with Hollerith punch code, a PRT100 and a DSS130 (connector number 29 in hexadecimal).

The following cards must be punched:

1. Order card

Col. 1 - 4 = 0396

Col. 5 - 6 = 06

Col. 7 - 10 = 0200

Col. 11 - 12 = b̸4

Col. 13 - 14 = 40

Col. 15 - 16 = b̸1

Col. 17 - 20 = blank

Col. 21 - 22 = 20

Col. 23 - 80 = blank

2. •SDG program detail cards

   a. RDC Card

      Col. 1 - 4 ƀRDC

      Col. 5 - 6 ƀ4

      Col. 7 - 9 064

      Col. 10 - 80 blank

   b. PRC Card

      Col. 1 - 4 ƀPRC

      Col. 5 - 6 ƀ1

      Col. 7 - 80 blank

   c. DSC Card

      Col. 1 - 4 ƀDSC

      Col. 5 - 6 ƀ1

      Col. 7 - 8 02

      Col. 9 - 11 032

      Col. 12 - 14 033

      Col. 15 - 80 blank

d. PNC Card

Col. 1 - 4 ƀPNC
Col. 5 - 6 ƀ1
Col. 7 - 9 064
Col. 10 - 80 blank

e. CDC Card

Col. 1 - 4 ƀCDC
Col. 5 - H
Col. 6 - 80 blank

f. MCC Card

Col. 1 - 4 ƀMCC
Col. 5 - 6 08
Col. 7 - 80 blank

g. END Card

Col. 1 - 4 ƀEND
Col. 5 - 80 blank

Cards must be feed into the card reader in the following order:

- Loader
- Load Order by Cards program with the parameters card preceded by the object program cards and immediately followed by the FF card
- •SDG program detail cards.

PART TWO DOS ORDERS

•DNA DISC NUMBER ASSIGNMENT

1. FUNCTION

To assign a logical disc number to a disc unit. These two numbers may or may not be the same.
A table relating logical disc number to physical disc units is set up in the memory at initial address 016E.

2. SYSTEM UNITS REQUIRED

The program uses the following peripherals:

- a disc unit for the System Disc
- a printer.

3. PARAMETER CARD

Disc Number Assignment requires only the order card with the following format:

Col. 1 to 4	Order •DNA identification code.
Col. 5 to 9	Not used.
Col. 10 to 19	Logical disc unit numbers. The logical number of the first unit is punched from column 10 on. Where a unit has the same logical/physical number, this column is not punched. An asterisk (x) is punched in the corresponding column for a disc assigned as out of order. Columns for unconnected disc units are left blank.
Col. 20 to 72	Not used.
Col. 73 to 80	Not used.

4. DESCRIPTION AND EXECUTION

The Disc Number Assignment program has only one segment which is stored at memory address 5024 (decimal) and is located on cylinder 0 of each disc.

The program creates a table relating logical disc unit numbers to physical disc units (this table starts at address 016E); i.e. the physical number associated with logical disc unit number 1 is located at address 016E, etc. The character FF signifies that a unit is not connected or is declared as out of order.

5. CHECKS AND ERROR MESSAGES

The Disc Number Assignment program checks the operation of peripheral units and the validity of parameters punched on order cards. The following messages are printed to indicate errors encountered:

0801 x D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or verifying on a logical disc unit N.
Program terminates; feasibility condition "NO".

0802 x D. UNIT N OUT OF ORDER

Unit N indicated as out of order. The program causes the printer to move the paper in the printer to a new page and the program halts at location 01DA.

Pressing RUN retries the operation that caused the signal. If the signal repeats, the program terminates. Feasibility condition "NO".

0803 x D. UNIT N PERSISTENT ERROR

Positioning error on unit N.
Program Terminates. Feasibility condition: "NO".

0810 * ORDER CARD ERROR

In addition to the asterisk, non-numeric punch in columns 10 to 19 of the order card.
Program terminates; feasibility condition: "NO".

0811 * MULTIPLE NUMBER

The same logical disc number has been assigned to more than one physical disc unit.
Program terminates: feasibility condition: "NO".

0812 * NOT DISC

A logical disc number has been assigned to an unconnected physical disc unit.
Program terminates; feasibility condition: "NO".

0813 * PROCEEDING IMPOSSIBLE INSERT PILOT ZERO

Message printed after each of the above error messages. It is a reminder that for an error in order •DNA execution, the system must be restarted.

6. EXAMPLE

For a system with 4 disc units connected, where the logical disc number 1 is to be assigned to the first disc unit (physical unit number 0) and the fourth unit (physical unit number 3) is to be ignored, the order card is punched as follows:

Col. 1 to 4	•DNA
Col. 5 to 9	b̸...b̸
Col. 10	1
Col. 11	0
Col. 12	b̸
Col. 13	*
Col. 14 to 80	b̸...b̸

•INT INITIALIZE

1. FUNCTION

To read the day's date into memory locations 40 to 45 (decimal) and data, which may be supplied by the user, in memory locations 52 to 83 (decimal) it sets the checking area (decimal locations 50 and 51) to zero (binary) and, if desired, prints and checks the serial numbers of volumes mounted on the disc units to be initialized.

2. SYSTEM UNITS REQUIRED

The following peripherals are required:

- a System Disc unit,
- the number of disc units (maximum of three) specified on the order card,
- a printer.

3. PARAMETER CARDS

The Initialize program requires only one order card with the following format:

Col. 1 to 4	order identification code: • INT
Col. 5 to 10	day's date in the form: ⌀YYDDD
Col. 11 to 16	serial number of the SD if checking required; blank if no checking required.

Col. 17	logical disc number of the first volume for which a label is to be written. Blank if no label checking is required. In all order cards having WD (Work Disc) punched in columns 5 and 6 reference will be made to the unit whose logical number is punched in this columns. The •APS order also refers to this unit each time a Work Disc is being used (option 3, card column 5).
Col. 18 to 23	Serial number of the volume specified in column 17. Blank if no serial number checking required.
Col. 24 to 30 Col. 31 to 37	Same as columns 17 - 23 if two disc units are to be initialized at the same time.
Col. 38	x for user data to be transferred to the portion of the memory reserved for this purpose. b̸ otherwise.
Col. 39 to 70	user data (optional); blank unless there is an asterisk x in card column 38.
Col. 71 to 80	not used.

4. DESCRIPTION AND EXECUTION

The Initialize program is made up of a single segment at address 2048 (decimal).
The card is first checked for validity, requested transfers are carried out, and the following information is printed:

- the message DISC LABEL OF UNIT N followed by the volume label of the SD,
- the message AVAILABLE CYLINDERS TABLE followed by the free area table of the SD,

- the message PROGRAM FILE LABEL, followed by the label,
- and, finally, the labels of other discs, if requested.

5. CHECKS AND ERROR MESSAGES

Messages are printed as a result of program checks. Note, however, that if there is an error during program execution processing cannot continue: a HLT stops the program at address 0000, followed by a jump at the same address; it is then necessary to reload Pilot 0 and restart the program.

The following is printed at the end of each message:

0318 x PROCEEDING IMPOSSIBLE - INSERT PILOT ZERO.

Error messages are as follows:

0301 x D. UNIT N PERSISTENT ERROR

Persistent error during reading on disc unit N.

0302 x D. UNIT N OUT OF ORDER

Disc Unit N is out of order. The program causes the printer to slew the paper to a new page and processing halts at location 01DA. Pressing RUN retries the operation that caused the message. If there is a new message, the program stops.

0303 x D. UNIT N PERSISTENT ERROR

Persistent error in positioning of disc unit N.

0310 x DATE CODE ERROR

Error in the card area containing the date (this might be a character other than ƀ in the first column; or an alphabetic character in the next columns).

0311 x D. UNIT N NUMBER ERROR

Unit N is not part of the system or is out of order.

0312 x USER DATA CODE ERROR

A symbol other than x or b̸ is punched in column 38 of the order card.

0317 x UNIT N DISC SERIAL NUMBER ERROR

The serial number punched on the order card does not agree with the serial number of the volume mounted on disc unit N.
The program prints the volume label on the next line.

6. EXAMPLE

Assuming that the serial number of the SD is to be verified as "1 2 3 4 5" that the labels of the two other volumes on disc units 0 and 4 are to be printed, that the serial number of the volume mounted on disc unit 0 is to be checked and that all user data areas are to be blank-filled, the order card should be punched as follows:

Col. 1 to 4	•INT
Col. 5 to 10	b̸ 6 7 1 5 5
Col. 11 to 16	b̸ 1 2 3 4 5
Col. 17	0
Col. 18 to 23	b̸23456
Col. 24	4
Col. 25 to 30	b̸ . . . b̸
Col. 31 to 37	b̸ . . . b̸
Col. 38	*
Col. 39 to 70	b̸ . . . b̸
Col. 71 to 80	b̸ . . . b̸

•IND INITIALIZE DISC

1. FUNCTION

To execute volume serialization by recording headers and reproducing cylinder zero from SD.

2. SYSTEM UNITS REQUIRED

The Initialize Disc requires the following peripherals:

- a System Disc unit
- a disc unit on which the volume to be serialized is mounted
- a printer.

3. PARAMETER CARD

The Initialize Disc requires only the order card with the following format:

Col. 1 to 4	Order identification code: •IND
Col. 5 to 10	Not used
Col. 11	Logical disc unit number on which volume to be serialized is mounted
Col. 12 to 17	Volume serial number
Col. 18 to 27	Volume name
Col. 28 to 72	Not used
Col. 73 to 80	Serial code (if desired).

4. DESCRIPTION AND EXECUTION

This program is made up of a single segment stored at memory location 2048 (decimal).

The sector headings of the entire volume are written sequentially from 00000 to 19999.

Cylinder zero is written in the same format as the one on the SD: sectors 00000 and 00001 are reserved for diagnostic programs, sector 00018 (physically the third sector contains the volume label, i.e. information punched on the order card; sector 00019 contains the free area table and designates cylinders 1 to 99 as free, except those containing a persistent error as revealed during the writing of headings; sector 00002 to 00017, reserved for the file labels, contains dummy labels with an indication of absent files (V in position 54); sectors 00020 to 00199 are duplicated from the corresponding sectors on the SD.

Then all sectors of cylinders 1 to 99 are written with a standard configuration of characters that are readable in both 8-bit and 6-bit mode.
Thus even though the file is read before being written, no disc unit error will occur.

Headings are written one cylinder at a time as follows:

- writing in "direct seek"
- verifying in "return to home seek"

It should be noted that only one volume can be serialized at a time. Consequently there must be one •IND card for each volume to be serialized.

5. CHECKS AND ERROR MESSAGES

The •IND (INITIALIZE DISC) program checks the operation of disc units and the validity of parameters punched on the order card.

Errors encountered are indicated by the following messages:

0201 ж D. UNIT N PERSISTENT ERROR

Persistent error in reading disc unit N.
Program terminates; feasibility condition: NO.

0202 ж D. UNIT N OUT OF ORDER

Disc Unit N is indicated as out of order.
The program causes the printer to move the paper to a new page and the program halts at location 01DA; pressing RUN retries the operation that caused the signal. If the signal repeats, the program terminates and the feasibility condition is NO.

0203 ж D. UNIT N PERSISTENT ERROR

Persistent positioning error on unit N.
Program terminates; feasibility condition: NO.

0210 ж D. UNIT N NUMBER ERROR

Error on the order card: column 22, containing the logical disc number of the unit to be serialized does not contain a numeric punch.
Program terminates; feasibility condition: NO.

0211 ж TRACK XXXXX WITH PERSISTENT ERROR CYLINDER YY NOT AVAILABLE

Persistent error in writing headings on the track whose first segment address is XXXXX. The program indicates on the free area table that the whole cylinder YY, on which this track is found, is not available. Program continues; feasibility condition: YES.

0212 x PERSISTENT ERROR ON CYLINDER 0

Persistent error on cylinder zero. Program terminates; feasibility condition: NO.

0213 x D. UNIT N NUMBER ERROR

The disc unit indicated in order card column 11 is not part of the systems.
Program terminates; feasibility condition: NO.

0214 x IMPOSSIBLE INITIALIZATION

Overflow of sector 00019 containing the free area table (in this case, during volume serialization, the number of blank cylinders was so great that the volume itself was divided into more than 24 free areas).
Program terminates; feasibility condition: NO.

6. EXAMPLE

To serialize a volume mounted on unit 1 with volume serial number 00012A and the user name O.G.E., the order card, which is not serially connected with other cards, should be punched as follows:

Col. 1 to 4	•IND
Col. 5 to 10	b̸ . . . b̸
Col. 11	1
Col. 12 to 17	00012A
Col. 18 to 27	O.G.E.
Col. 28 to 72	b̸ . . . b̸
Col. 73 to 80	b̸ . . . b̸

•IDC INITIALIZE DISC by CARD

1. FUNCTION

This program initializes a volume, i.e. writes all disc sector headers and creates the cylinder zero, using the informations read from a data card deck.

2. SYSTEM UNITS REQUIRED

This program uses the following peripherals:

- a disc unit,on which are mounted the SD, in the first phase; the volume being initialize, in the second phase
- a card reader, to read the data card deck
- a printer
- a punch, if in the first phase the data card deck is to be punched.

3. PARAMETER CARDS

This program requires only the order card, with the following format:

Col. 1 to 4	•IDC
Col. 5 to 10	not used
Col. 11	P, if in the first phase, the data card deck is to be punched, otherwise blank
Col. 12 to 17	serial number of the volume
Col. 18 to 27	user name
Col. 28 to 72	not used
Col. 73 to 76	optional ID code
Col. 77 to 80	not used

4. DESCRIPTION AND EXECUTION

This program is made up of a single segment with start address 1800 (decimal). Program execution is made up of two successive phases:

- Initially, after the System Disc is mounted, the program is loaded and its execution is started by Pilot 2. If option P is present in col. 11 of the order card, the contents of Cylinder Zero of the System Disc, with a structure similar to that of a program punched with •APP, are punched on cards.
  BPbb (Basic Program) code is punched in the first four columns of these cards.
  At the end, the program halts after printing message 1817.

- In the second phase, after substituting the SD with the volume being initialized, the card reader is loaded with the data card deck and the initialization phase begins.

  The volume is made available for writing in sector mode. The sector headings of the entire volume are written sequentially from sector 00000 to 19999. Cylinder Zero is written in the following way.

sectors 00000 - 00001	are reserved for diagnostic programs
sector 00018	(physically the 3rd sector) contains the volume label, using informations punched on the order card.
sector 00019	contains the free area table and designate cylinders 1 to 99 as free, except those containing a persistent error as revealed during the writing of heading
sector 00002 to 00017	reserved for the file labels; containing dummy labels with an indication of absent files (V in position 54)
sector 00020 to 00199	are written, using the data card deck, as the corresponding sectors on the SD

Then all sectors of cylinders 1 to 99 are written with a standard configuration of characters that are readable in both 8-bit and 6-bit mode. Thus even though the file is read before being written, no disc unit error will occur.

It should be noted that only one volume can be serialized at a time. Consequently there must be one •IDC card for each volume to be serialized.

5. CHECKS AND ERROR MESSAGES

Errors encountered are indicated by the following messages:

1801 * D. UNIT N PERSISTENT ERROR

Persistent error in reading disc unit N.
Program terminates; feasibility condition: NO.

1802 * D. UNIT N OUT OF ORDER

Disc Unit N is indicated as out of order.
The program causes the printer to move the paper to a new page and the program halts at location 01DA; pressing RUN retries the operation that caused the signal . If the signal repeats, the program terminates and the feasibility condition is NO.

1803 * D. UNIT N PERSISTENT ERROR

Persistent positioning error on unit N.
Program terminates; feasibility condition: NO.

1810 * ORDER CARD ERROR

A wrong punch is present in column 11 of the order card. Program execution terminates; feasibility condition: NO.

1811 * TRACK XXXXX WITH PERSISTENT ERROR CYLINDER YY NOT AVAILABLE

Persistent error in writing headings on the track whose first sector address is XXXXX. The program indicates on the free area table that the whole cylinder YY, on which this track is found, is not available. Program continues; feasibility condition: YES.

1812 * PERSISTENT ERROR ON CYLINDER 0

Persistent error on cylinder zero. Program terminates; feasibility condition: NO.

1814 * IMPOSSIBLE INITIALIZATION

Overflow of sector 00019 containing the free area table (in this case, during volume serialization, the number of blank cylinders was so great that the volume itself was divided into more than 24 free areas).
Program terminates; feasibility condition: NO.

1815 * NO PUNCHER

No puncher is connected to the system when P character is punched in col. 11. Program execution terminates; feasibility condition:NO.

1816 * PUNCH ERROR ON CARD **

Punch error on the card which is printed with two asterisk aside. Program execution goes on; feasibility condition: YES.

1817 * REPLACE S.D. DECK ON CARD READER

Program HLT 01CC signals the end of the first phase. At this point the user should substitute the SD with the volume to be initialized and load the data card deck on the card reader. By pressing START the program re-continues; feasibility condition: NO.

1818 * CARD ADDRESS ERROR

The memory address punched on the last card read does not correspond to the sum of the address plus the length punched on the preceding card. Program halts; feasibility condition: NO.

1819 * CARD SEQUENCE ERROR

The last card read is out of sequence. Program execution halts; feasibility condition: NO.

1820 * PROGRAM CODE ERROR

The program code punched in the first four columns of the last card read is not BPƀƀ; program execution halts; feasibility condition: NO.

1899 * END OF INITIALIZE DISC BY CARD
End of program

6. EXAMPLE

To initialize a volume, writing serial number 012345 and the user name HONEYWELL, using a data card deck already available at the Center.

The contents of the order card is the following

Col. 1 to 4	•IDC
Col. 5 to 11	blank
Col. 12 to 17	012345
Col. 18 to 27	HONEYWELL.
Col. 28 to 80	Blank.

•FLW FILE LABEL WRITE

1. FUNCTION

To insert new file labels (data or programs) on disc while preparing reserved volume areas of the file for future use; to delete or update file labels included in specific volumes and to alter the area reserved for these volumes at the same time.

2. SYSTEM UNITS REQUIRED

The File Label Write program uses the following peripherals:

- one or two disc units to write or modify file labels or delete existing files
- a System Disc unit
- a printer
- a card reader (to insert detail cards).

3. PARAMETER CARDS

3.1. ORDER CARD

The format of the order card is:

Col. 1 to 4	Order identification code: •FLW
Col. 5 to 72	Not used
Col. 73 to 80	Serial code (if desired).

3.2. DETAIL CARDS

To insert a new label into the file table, the detail card FLI is required; to update the file table, the detail card FLU is required; to delete a label from the file table, the detail card FLD is required.

3.2.1. FLI card (File Label Insert) for insertion of new labels.

The format of this card is:

Col. 1 to 4	Identification code: ƀ F L I
Col. 5 to 6	Logical disc number, punched in column 6 of the unit on which the volume concerned is mounted; punch SD for a System Disc.
Col. 7 to 12	Volume serial number
Col. 13	Type of file - S = Sequential - R = Random - P = Program
Col. 14 to 25	File name - if type P columns 14 - 15 must contain SD for a System Disc, or MD for a Master Disc, or LD for a Library Disc.
Col. 26 to 31	Serial number of the first volume of the file.
Col. 32 to 35	Sequence number of the volume within the file.
Col. 36	File code indicating volume location within the file. This may have one of the following values: - F = last volume in the file - V = file continues on other volumes.

Col. 37	Recording mode - 0 : 6-bit sector mode - 1 : 8-bit sector mode - 2 : 6-bit full-track mode - 3 : 8-bit full-track mode
Col. 38 - 43	File creation date; if blank, the day's date located in the memory is used by the program.
Col. 44 to 49	Retention date; if blank, creation date is used.
Col. 50 to 51	Initial cylinder of the first part of file (if the letter A is punched in card column 50, the program selects the cylinder according to the free area table).
Col. 52 to 53	Number of cylinders to be reserved for the first part of file.
Col. 54 to 57	Same as columns 50 to 53 for second part of file (if required). It should be noted that these columns remain blank for program files or where the choice of file area is requested from the program. In the latter case, any division of the file into two parts is carried out by the program if there is no area large enough to contain the whole file.
Col. 58 to 59	Initial cylinder of overflow area (if the letter A is punched in column 58, the overflow area is selected by the program). This parameter is only valid for type S files.
Col. 60 to 61	Number of cylinders for overflow area; this parameter is only valid for type S files.
Col. 62 to 63	Cylinder for index; this parameter is only valid for type S files. If the letter A is punched in column 62, the cylinder is selected by the program.

Col. 64 to 80	Not used. It should be noted that if automatic selection of file areas is requested, all other areas must be selected automatically.

3.2.2. F L U Card (File Label Update) to update an existing label

The format of an FLU detail card is as follows:

Col. 1 to 4	Identification code: b̸ F L U	
Col. 5 to 6	Logical disc number, punched in column 6, on which the volume containing the label to be modified is located, punch SD if the volume is a System Disc.	
Col. 7 to 12	Volume serial number.	
Col. 13	Type of file on which the label to be modified is located.	
Col. 14 to 25	File name of label being updated.	
Col. 26 to 49	These columns correspond to the field breakdowns of the FLI cards and contain new data to be inserted in new labels. Leave any field not to be updated blank. Modification of the recording mode (Col. 37) is not permitted.	
Col. 50	- x	To reduce the reserved area of the file to the dimensions it actually takes up. Not valid for the overflow area of a sequential file or a random file.
	- b̸	Otherwise.
Col. 51	Blank	

Col. 52 to 53	Number of cylinders to be assigned to the file; this number may be greater or smaller than the one already assigned to the file. If column 50 contains an asterisk x, this area is left blank. Left blank, as well, if there is no change in the number of cylinders reserved for the file.
Col. 54	In increasing the number of cylinders reserved for the file, it may be necessary to create a second area. If an asterisk x is punched in this columns, the program automatically selects the second part of the file. Blank, if the file is not to be divided into two parts.
Col. 55 to 57	Not used.
Col. 58 to 59	Number of the initial cylinder for overflow area to be created (if necessary).
Col. 60 to 61	Number of cylinders assigned to the overflow area, whether for updating or creation.
Col. 62 to 63	Cylinder number to be reserved for a type S file index, if necessary.
Col. 64 to 75	New file name (if desired).
Col. 76 to 80	Not used.

3.2.3. F L D Card (File Label Delete) to delete an existing label

The format of the detail card used to delete an area previously reserved for a file in the volume is as follows:

Col. 1 to 4	Identification code: ƀ F L D
Col. 5 to 6	Logical disc number, punched in column 6, on which the volume containing the label to be deleted is located; punch SD if the volume is a System Disc.

Col. 7 to 12	Volume serial number.
Col. 13	Type of file to be deleted (S, R or P) * if all the files on the volume are to be deleted; if columns 5 and 6 contain SD, all the files are deleted from the System Disc except the program file.
Col. 14 to 25	Name of the file to be deleted (if there is no asterisk in column 13).
Col. 26 to 80	Not used.

3.2.4. End of order card

A ꞓEND card must be placed at the end of all detail cards.

4. DESCRIPTION AND EXECUTION

The program is made up of four segments stored at location 1800 (decimal).

To write a new label, the appropriate area is sought and the free area table is consulted; the new label is then created and written. To update or delete a label, a check must first be made to be certain that this label exists.

The program prints the new file labels to be written, modified or deleted.

To delete a program file, location 53 of the corresponding label contains the character V, which indicates that the file is deleted and that the locations are now free.

To write a new file, the program writes the new label in the sectors reserved for this purpose on the first track of cylinder zero; the free area table is updated and, for files to be written in the full-track mode, headings of those tracks reserved for the file are written. To write an R type file, the program writes an "A0" (hexadecimal) end of page code at the beginning of all usable pages. (By usable pages is meant file sectors when writing in the sector mode and file tracks when writing in the full-track mode).

The program reacts in the same way to modify the dimensions of an existing file.

To summarize, the program fulfills the following functions

a) creation of new file labels and preparation of volume areas

b) deletion of the volume of certain files or of complete existing files

c) deletion of the SD for all except program files

d) modification of the size of the volume area reserved for a specific file or for its overflow area

e) modification of one or more parameters on the HDR1 label

f) addition or deletion of the overflow area or of an index cylinder.
In addition, the program will always carry out the following operations:

- if labels have been deleted, it rearranges the valid labels so that they are in the first area reserved for them,
- if changes have occured in the free areas, the corresponding table is updated so that the first area indicated is always the largest.

5. CHECKS AND ERROR MESSAGES

The program not only checks the peripheral units but also checks the order card and detail card areas, both physically and logically.

When an error is found, one of the following messages is transmitted:

0701 x D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or verifying logical disc unit N. Program terminates; feasibility condition: NO.

0702 x D. UNIT N OUT OF ORDER

Disc unit N signalled out of order. The program causes the printer to move the paper to a new page and the program halts at location 01DA; pressing RUN retries the operation that caused the message. If the signal repeats, the program terminates. Feasibility condition: NO.

0703 x D. UNIT N PERSISTENT ERROR

Positioning error on unit N.
Program terminates; feasibility condition: NO.

0710 x SUB-ORDER CARD ERROR

Wrong identification code on card read.
Program terminates; feasibility condition: NO.

0711 x SUB-ORDER CARD COLUMN N ERROR

The area beginning with column N of the previously printed detail card is wrong.
Program terminates; feasibility condition: NO.

0712 x D. UNIT N NUMBER ERROR

System does not include a logical disc unit with the number indicated.
Program terminates; feasibility condition: NO.

0713 x SERIAL NUMBER ERROR XXXXXX

The volume serial number, as punched in card columns 7 to 12, does not agree with the serial number actually written on the volume.
Program terminates; feasibility condition: NO.

0714 x PERSISTENT ERROR CYLINDER YY

Persistent error in writing headings on cylinder number YY.
Program terminates; feasibility condition: NO.

0715 x NOT EXECUTABLE COLUMN NN

The area beginning with columns NN cannot be executed. (e. g. a file increase was requested but the available area is not great enough to contain it).
Program terminates; feasibility condition: NO.

0716 x FILE UNKNOWN

Volume indicated does not contain a file with the name punched in columns 14 to 25 of the FLU or FLD cards.
Program terminates; feasibility condition: NO.

0717 x OVERFLOW FILE LABEL

No further file labels can be written on sectors 00002 to 00017 of cylinder zero.
Program terminates; feasibility condition: NO.

0718 x OVERFLOW AVAILABLE CYLINDERS TABLE

Overflow beginning with segment 00019 containing the free area table. Indicates the maximum limit or more then 24 areas have been created.
Program terminates; feasibility condition: NO.

0719 x NOT EXECUTABLE

General message indicating that an order cannot be executed.
The program prints this message when it cannot give more precise details as in 0715.
Program terminates; feasibility condition: NO.

7. EXAMPLE

Assuming all "data" files are to be deleted from the System Disc, the order card should have the following format:

Col. 1 to 4	•FLW
Col. 5 to 80	␢ . . . ␢

The detail card following the order card should be punched as follows:

Col. 1 to 4	␢FLD
Col. 5 to 6	SD
Col. 7 to 12	OGE 0001
Col. 13	*
Col. 14 to 80	␢ . . . ␢

An END card must follow the detail card.

Col. 1 to 4	␢END
Col. 5 to 80	Blank.

•APS ASSEMBLE APS PROGRAM

1. FUNCTION

This Order calls out the APS Assembly program for translating from source to machine language, assembling the subroutine called in turn, if any.

2. SYSTEM UNITS REQUIRED

Assemble APS Program requires the following peripherals

- a Disc unit
- a printer
- a card reader

3. PARAMETER CARDS

This program requires only the order card with the following format:

Col. 1 to 4	•APS
Col. 5	source program medium: ƀ, program is punched on cards codified in Basic APS or in APS II 1, program is punched on cards codified in Extended APS 2, program resides on free area of System Disc and the SD is the only medium used. This is a typical case with TAB III and SPG 3, program is punched on cards and a free area of the work disc is utilized 4, program resides on the free area of the SD but a work disc is used for the assembly

Col. 6	source program: b̸, if it is a program 1, if it is a subroutine 2, if it is an EIOS subroutine (Discs)
Col. 7 to 14	name of program or subroutine
Col. 15 to 49	program description printed as the heading of the object program listing
Col. 50 to 72	not used
Col. 73 to 76	optional ID code
Col. 77 to 80	not used.

4. CHECKS AND ERROR MESSAGES

For error controls and consequent messages see APS manual.

•COB COMPILE COBOL PROGRAM

1. FUNCTION

This Order calls COBOL Compiler for translating a program from source to machine language.

2. SYSTEM UNITS REQUIRED

This program, with a minimum of 12 K store, requires the following peripherals:

- a card reader
- a printer
- a Disc unit for the SD which must have 48 free cylinders.

3. PARAMETER CARDS

This program requires only the order card whose contents are the following:

Col. 1 to 4	•COB
Col. 5 to 8	Not used
Col. 9	D
Col. 10	L, program listing is to be output ƀ, program listing is not to be output
Col. 11	ƀ, the source program is to be compiled from cards D, the source program is to be compiled from disc
Col. 12	Not used

Col. 13 to 20	if D in column 11, the name of the program to be compiled, otherwise not used
Col. 21 to 25	Object program decimal start address (optional)
Col. 26 to 37	Not used
Col. 38 to 39	DM, "Debug Mode" is requested. b̸b̸, 'Debug Mode' is not requested
Col. 40 to 71	Not used
Col. 72 b	b̸, if source program is on card. Number of the Disc unit on which the source program is stored.
Col. 73 to 76	Optional ID code
Col. 77 to 80	Not used

4. CHECKS AND ERROR MESSAGES

For error controls and consequent messages see COBOL Reference Manual.

•FOR COMPILE FORTRAN PROGRAM

1. FUNCTION

This Order calls FORTRAN Compiler for Translating a program from source to machine language.

2. SYSTEM UNITS REQUIRED

This program, with a minimum of 12 K store, requires the following peripherals:

- a card reader
- a printer
- a Disc unit

3. PARAMETER CARDS

3.1. ORDER CARD

Col. 1 to 4	•FOR
Col. 5 to 8	Not used
Col. 9	D
Col. 10	Kind of medium containing the source program: D = disc ƀ, or any character = cards
Col. 11	Option for the printout of the object program L = program listing is to be output ƀ = program listing is not to be output
Col. 12	W, if a Work Disc is assigned in addition to the System Disc ƀ, if only the System Disc is used

Col. 13	Memory size used when compiling the program 3 = 12 K store 4 = 16 K store
Col. 14	Memory size used when executing the program 3 = 12 K store 4 = 16 K store
Col. 15	b̸, if the object program does not use tapes 1, if the object program uses tapes
Col. 16	b̸, if the object program does not use discs 1, if the object program uses discs
Col. 17	b̸, if the object program does not use the card punch 1, if the object program uses the card punch
Col. 18 to 20	
Col. 21 to 25	Start address decimal of the object program; blank if the start address is 1800
Col. 26 to 33	program name
Col. 34 to 72	Not used
Col. 73 to 76	Optional ID code
Col. 77 to 80	Not used

3.2. END CARD

The source program (if punched on cards) is to be placed following •FOR card. In this the end of source program is signalled by END card with the following format:

Col. 1 to 4	b̸END
Col. 5 to 80	not used

4. CHECKS AND ERROR MESSAGES

For error controls and consequent messages, see FORTRAN Language manual.

•SGT GENERATE SORT/MERGE FOR TAPE

1. FUNCTION

This Order calls SGT compiler which provides to generate a Sort or Merge program for tape in APS language; this program is translated into basic language by means of the APS compiler.

2. SYSTEM UNITS REQUIRED

This Order requires the following peripherals

- a card reader
- a printer
- a disc unit

3. PARAMETER CARDS

This Order Requires only the order card whose contents are the following:

Col. 1 to 4	•SGT
Col. 5 to 8	Not used
Col. 7 to 14	name of the program which is compiled
Col. 15 to 49	program description, which is printed as header of the program listing
Col. 50 to 72	Not used

Col. 73 to 76	Optional ID code
Col. 77 to 80	Not used

4. CHECKS AND ERROR MESSAGES

For error controls and consequent messages see the manual Sort Generator for Tape.

•SPG GENERATE SORT/MERGE PROGRAM

1. FUNCTION

This Order calls SPG compiler which provides to generate a Sort or Merge program in APS language; this program is translated into basic language by means of the APS compiler.

2. SYSTEM UNITS REQUIRED

This Order requires the following peripherals:

- a card reader
- a printer
- a Disc unit

3. PARAMETER CARDS

This Order requires only the order card whose contents are the following:

Col. 1 to 4	•SPG
Col. 5 to 8	Not used
Col. 7 to 14	name of the program which is compiled
Col. 15 to 49	program description, which is printed as header of the program list
Col. 50 to 72	Not used
Col. 73 to 76	Optional ID code
Col. 77 to 80	Not used

4. CHECKS AND ERROR MESSAGES

For error controls and consequent messages see the manual Sort Program Generator.

•TAB TRANSLATE TAB PROGRAM

1. FUNCTION

This Order calls TAB compiler which provides to translating a program written in TAB language into APS; this program is then translated by mean of APS compiler.

2. SYSTEM UNITS REQUIRED

This Order requires the following peripherals:

- a card reader
- a printer
- disc unit

3. PARAMETER CARDS

3.1. ORDER CARD

Col.	1 to 4	•TAB
Col.	5	ƀ, if a Work Disc is assigned in addition to the System Disc. 1, if only the System Disc is used. 2, if only TAB compilation is desired.
Col.	6	ƀ, if the source program is written in TAB III 1, if the source program is written in TAB I or TAB II
Col.	7 to 14	program name; first four characters must be equal to the program code punched in the first four columns of the source deck.

Col. 15 to 34	program description, which is printed as header of the program list.
Col. 35 to 72	Not used.
Col. 73 to 76	Optional ID code
Col. 77 to 80	Not used.

3.2. END CARD

After •TAB card must be placed source program cards whose end is signalled by END card:

Col. 1 to 4	ƀEND
Col. 5 to 80	Not used.

4. CHECKS AND ERROR MESSAGES

For error controls and consequent messages see the manuals relative to the TAB language.

•API ASSEMBLED PROGRAM INSERT

1. FUNCTION

To insert an object program, generated in the work area of a compiler, into the program file of the system disc. This is the first and largest area indicated in the free area table of the SD.

To insert a new program, the file must be reorganized: all type 3 programs generated for controlled execution are deleted and the new program is inserted in the file following those remaining.

Following the program are printed the volume label, the program file update label and the message:

DISC LABEL AND FILE LABEL

It should be noted that only one program is to be written at a time and that the order •API must come immediately after the compilation order •APS.

2. SYSTEM UNITS REQUIRED

The following peripherals are needed:

- a disc unit on which the System Disc is mounted
- a printer.

3. PARAMETER CARDS

The Assembled Program Insert used only the order card with the following format:

Col. 1 to 4	Order identification code: •API
Col. 5	Program Type
Col. 6 to 13	Program Name
Col. 73 to 80	Serial code (if desired)

4. DESCRIPTION AND EXECUTION

The Assembled Program Insert is made up of a single segment located at address 2048 (decimal).

The object program is inserted in the work area in the 6-bit mode and organized in pages having a maximum of 512 characters in unpacked form. •API writes the program in the program file in 8-bit mode and organizes it in pages having a maximum size of 1024 characters.

5. CHECKS AND ERROR MESSAGES

Checks carried out by the program on peripheral units and to determine the physical precision of the order card, may give rise to the following error messages:

1001 x D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or verifying logical disc unit N.
Program terminates, feasibility condition: NO.

1002 x D. UNIT N OUT OF ORDER

Unit N signalled out of order. The program causes the printer to move the paper to a new page and halts at location 01DA; pressing RUN retries the operation that caused the Signal. If the signal repeats, the program terminates and the feasibility condition is: NO.

1003 x D. UNIT N PERSISTENT ERROR

Persistent positioning error on unit N. The program terminates; feasibility condition: NO.

1010 x MULTIPLE PROGRAM NAME

A program with the same name as the one being inserted exists on the program file.
The new program is still inserted and the old program must be deleted by means of an order •SDS, if the new program is to replace the old one.
At the end of the program, the feasibility condition is: NO.

1011 x OVERFLOW OF PROGRAM FILE TN. . . N LAST PROGRAM

The program file has surpassed the maximum dimension after writing a new program. File writing terminates, the last fully written program NN is a type T program. The program terminates; feasibility condition: NO.

1012 x PROGRAM CODE ERROR

Wrong code for the type of program to be written; column 5 of the order card does not contain a 1 or a 3.
Program terminates; feasibility condition: NO.

1013 x PROGRAM NOT AVAILABLE

The program written in the compiler working area is not the one indicated on •API order card.
Program terminates; feasibility condition: NO.

1014 x OVERFLOW OF PROGRAM TABLE

The program file index has reached the maximum size of 259 programs so that no other records can be written.
Program terminates; feasibility condition: NO.

1098 x PROGRAM FILE NOT AVAILABLE

As a consequence of the last error printed, the program file is no longer usable.

6. EXAMPLE

To insert a type 3 program named CANE, which has just been compiled at the end of the System Disc program file, the order card should be punched as follows:

Col.	1 to 4	•API
Col.	5	3
Col.	6 to 9	CANE
Col.	10 to 80	b̸ . . . b̸

•PRL PROGRAM LOAD

1. FUNCTION

To seek a type 3 program on the System Disc, Library Disc or Master Disc, load it and start it.

2. SYSTEM UNITS REQUIRED

The Program Load uses the following peripherals:

- a disc unit for the SD, LD or MD
- a card reader for detail card input
- a printer.

3. PARAMETER CARD

The program uses an order card as the parameter card plus LBC cards (see Appendix A), followed by an END card.

The order card has the following format:

Col. 1 to 4	Identification code: •PRL
Col. 5 to 12	Name of the program to be started.
Col. 13	C – if the program to be started is to be carried out in controlled form (using order •DPG). Otherwise left blank.

Col. 14	P - if the program is to be carried out using a trace routine, i.e. with printing of the jump instructions given. Otherwise left blank.
Col. 15 and 16	Not used.
Col. 17	Code explaining how data contained in columns 18 to 49 of the order card is to be used: x if the contents are to be moved into the user area. Blank otherwise.
Col. 18 to 49	Data to be moved into the user area (optional).
Col. 50 to 72	Not used.
Col. 73 to 80	Serial code, if any.

The END card and •PRL card must be present even if there are no LBC cards.

4. DESCRIPTION AND EXECUTION

The program is on the System Disc and is made up of 4 segments located respectively at memory addresses: 5024, 7872, 11 968, 16 064 (decimal). The 2nd, 3rd, and 4th segments contain the trace routine for 8K, 12K, 16K memories respectively.

The first segment searches for the check point file which has the label CHKP followed by the name of the program to be started. If not found, it searches the general CHKP file. If found, the initial address of the file (first sector of the first cylinder containing the file) is read into memory. For a negative search, the value 00000 is stored (decimal).

In addition the program makes sure that the check point size is twice the memory size.

After this, the program index is read and the requested program is retrieved.
If the program uses a file table, LBC cards are read and the file table is updated, the memory locations that are not used by the program are filled with characters 0A and the first segment is read into the memory and started.

If the trace routine is requested, a decision is made according to whether or not the available memory can accommodate the trace routine segment corresponding to the memory size that is being used. The •PRL program generated by the order •MDG and inserted in the MD also Functions as a Monitor. Indeed the Monitor provides all the Functions of the •PRL program once it is started by the appropriate order card.

5. CHECKS AND ERROR MESSAGES

The program checks the operation of the units and the validity of parameters punched in the order card.

Any error causes one of the following messages to appear:

1401 x D. UNIT N PERSISTENT ERROR

Persistent error in reading logical disc unit N containing program file. The program stops; feasibility condition: NO.

1402 x D. UNIT N OUT OF ORDER

Logical disc unit N containing the program file is out of order. The program slews the paper to the next page and halts at location 01DA. Pressing RUN retries the operation; if the message is repeated the program stops; feasibility condition: NO.

1403 x D. UNIT N PERSISTENT ERROR

Persistent positioning error on logical disc unit N containing the program file. The program stops; feasibility condition: NO.

Col. 14	P - if the program is to be carried out using a trace routine, i. e. with printing of the jump instructions given. Otherwise left blank.
Col. 15 and 16	Not used.
Col. 17	Code explaining how data contained in columns 18 to 49 of the order card is to be used: x if the contents are to be moved into the user area. Blank otherwise.
Col. 18 to 49	Data to be moved into the user area (optional).
Col. 50 to 72	Not used.
Col. 73 to 80	Serial code, if any.

The END card and •PRL card must be present even if there are no LBC cards.

4. DESCRIPTION AND EXECUTION

The program is on the System Disc and is made up of 4 segments located respectively at memory addresses: 5024, 7872, 11 968, 16 064 (decimal). The 2nd, 3rd, and 4th segments contain the trace routine for 8K, 12K, 16K memories respectively.

The first segment searches for the check point file which has the label CHKP followed by the name of the program to be started. If not found, it searches the general CHKP file. If found, the initial address of the file (first sector of the first cylinder containing the file) is read into memory. For a negative search, the value 00000 is stored (decimal).

In addition the program makes sure that the check point size is twice the memory size.

After this, the program index is read and the requested program is retrieved.
If the program uses a file table, LBC cards are read and the file table is updated, the memory locations that are not used by the program are filled with characters 0A and the first segment is read into the memory and started.

If the trace routine is requested, a decision is made according to whether or not the available memory can accommodate the trace routine segment corresponding to the memory size that is being used. The •PRL program generated by the order •MDG and inserted in the MD also Functions as a Monitor. Indeed the Monitor provides all the Functions of the •PRL program once it is started by the appropriate order card.

5. CHECKS AND ERROR MESSAGES

The program checks the operation of the units and the validity of parameters punched in the order card.

Any error causes one of the following messages to appear:

1401 x D. UNIT N PERSISTENT ERROR

> Persistent error in reading logical disc unit N containing program file. The program stops; feasibility condition: NO.

1402 x D. UNIT N OUT OF ORDER

> Logical disc unit N containing the program file is out of order. The program slews the paper to the next page and halts at location 01DA. Pressing RUN retries the operation; if the message is repeated the program stops; feasibility condition: NO.

1403 x D. UNIT N PERSISTENT ERROR

> Persistent positioning error on logical disc unit N containing the program file. The program stops; feasibility condition: NO.

1410 x PROGRAM NOT FOUND

The requested program has not been found in the program file. The program stops; feasibility condition: NO.

1411 x SUB ORDER CARD ERROR

The detail card read cannot be identified. The program stops; feasibility condition: NO.

1412 x SUB ORDER CARD COLUMN NN ERROR

Detail card area starting from column NN incorrect. The program stops; feasibility condition: NO.

1413 x EXECUTION WITH PATH DUMP NOT POSSIBLE

No available area in the memory to load the trace routine as requested.
The program is started nonetheless; feasibility condition: YES.

1414 * FILES-TABLE ADDRESS ERROR

The start address of a program containing the files-table, is greater than 0FA0, and it is not possible updating the files-table. Program execution stops. Feasibility condition:NO.

1415 x ORDER CARD COLUMN NN ERROR

Card •PRL area starting from column NN incorrect.
The program stops; feasibility condition: NO.

1416 x CHECK POINT FILE TOO SMALL

The volume area reserved for the check point file is not of sufficient lenght. This message also appears when the program does not provide for the use of this file. The program stops; feasibility condition: NO.

6. EXAMPLE

To load and start a program with the code 5002 that has been generated by the order ●DPG on a card with the serial code 0003, the order card should be punched as follows:

Col. 1 to 4	●PRL
Col. 5 to 12	5002
Col. 13	C
Col. 14 to 16	b̸ . . . b̸
Col. 17 to 72	Not used
Col. 73 to 80	0003

This card is followed by the END card:

Col. 1 to 4	b̸END
Col. 5 to 80	Blank.

•APP ASSEMBLED PROGRAM PUNCH

1. FUNCTION

To punch a program that has just been translated or a program from the SD or LD. As the program thus obtained can be used immediately by the •SDU and •LPC programs, PLC and SIN cards are also punched.

2. SYSTEM UNITS REQUIRED

Assembled Program Punch requires the following peripherals:

- a disc unit on which the SD or LD is mounted
- a printer
- a card punch

3. PARAMETER CARD

Assembled Program Punch requires only the order card with the following format:

Col. 1 - 4	Order identification code: •APP
Col. 5	Type of program to be punched (1 or 3)
Col. 6 - 13	Program name
Col. 14	ƀ If the program to be punched is already included in the program file * If the program has just been translated so that it is located in the service area.

Col. 15 to 18	Code of the program to be punched in columns 1 to 4 of the program card.
Col. 19	ƀ if column 36 of the PLC card is punched with an I; R if column 36 of the PLC card is punched with an R.
Col. 20	Code specifying if the (decimal) number of cards for each program segment is to be punched in columns 33 - 36 of the SIN and END cards: - any code if punching requested, - otherwise left blank.
Col. 21 to 72	Not used.
Col. 73 to 80	Identification code.

4. DESCRIPTION AND EXECUTION

The Assembled Program Punch is made up of a single segment located at address 2048 in the memory. It contains the various card punch subroutines for the various types of punches; subroutine selection is based on indications contained in the DOS area reserved for system configuration.

The program prints all the cards punched and, if a punching error occurs, prints an asterisk (x) opposite the contents of the card involved. The punched detail cards PLC, SIN and END will have the date punched as ƀYYDDD in columns 37 - 42

5. CHECKS AND ERROR MESSAGES

The Assembled Program Punch checks the units and the validity of the parameter card; one of the following messages is printed when an error is encountered:

2101 x D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or verifying logical disc N. The program stops; feasibility condition: NO.

2102 x D. UNIT N OUT OF ORDER

Logical disc N is out of order. The program causes the printer to slew to the next page and halts at location 01DA.
Pressing RUN retries the operation that caused the message.
If the message repeats, the program stops; feasibility condition: NO.

2103 x D. UNIT N PERSISTENT ERROR

Persistent positioning error on logical disc N.
The program stops; feasibility condition: NO.

2110 x ORDER CARD ERROR

An error has been found on the order card.
The program stops; feasibility condition: NO.

2111 x PROGRAM NOT FOUND

The requested program cannot be found in the file or the service area. The program stops; feasibility condition: NO.

2112 x PROGRAM NOT AVAILABLE

The program in the service area is not the same as the one mentioned on the order card. The program stops; feasibility condition: NO.

2113 x PROGRAM NOT AVAILABLE

The program in the service area is not the same as the one mentioned on the order card. The program stops; feasibility condition: NO.

6. EXAMPLE

To punch a program named FASE 1 located in the SD with the identification code 4047 the order card should be punched as follows:

Col. 1 to 4	●APP
Col. 5	3
Col. 6 to 13	FASE 1
Col. 14	ƀ
Col. 15 to 18	4047
Col. 19	R
Col. 20 to 80	Blank.

•PCC PUNCHED CARD CHECK

1. FUNCTION

This Order allows to control the object program cards punched with Order •APP and the object program written on the program file of the SD or on the work area.

2. SYSTEM UNITS REQUIRED

This program requires the following peripherals:

- a card reader
- a printer
- a Disc unit for the SD

3. PARAMETER CARDS

This program uses the order card only, which is followed by the card deck punched by the •APP program. The order card has the following format:

Col. 1 to 4	•PCC
Col. 5 to 13	type code and name of the program to be controlled
Col. 14	ƀ, if the program to be controlled is written on the program file of the SD *, if the program to be controlled is written on the work area
Col. 15 to 18	program code punched on the object cards in columns 1 ÷ 4

Col. 19 to 72	Not used
Col. 73 to 76	Optional ID code
Col. 77 to 80	Not used.

4. DESCRIPTION AND EXECUTION

The program consists of a single segment. First the validity of the order card parameters is checked, then if no error occurs the object program cards are read and compared with the disc data.

If there is any difference, the card in question is printed, preceded by two asterisk as well as the corresponding part of the disc.

5. CHECKS AND ERROR MESSAGES

The program makes the following checks:

- the validity of the codes punched on the order card
- the presence of the program to be checked on the program file or on work area of the SD, as indicated in col. 14 of the order card
- the validity of cards punched.

The following messages may be printed:

2401 * D. UNIT N PERSISTENT ERROR

Persistent read error on disc unit N. The program halts at address 02A6. Feasibility condition: NO.

2402 * D. UNIT N OUT OF ORDER

Logical disc unit N out of order. The program causes the printer to move the paper to a new page and halts at location 01DA. Pressing START retries the operation that caused the signal. If the signal repeats, the program terminates; feasibility condition NO.

2403 * D. UNIT N PERSISTENT ERROR

Persistent positioning error on logical disc unit N.
Program stops; feasibility condition: NO.

2410 * ORDER CARD ERROR

Wrong parameters on the order card;
Program execution halts; feasibility condition: NO.

2411 * PROGRAM NOT FOUND

The program to be checked is not present either in the program file nor in the work area.
Program execution halts; feasibility condition: NO.

2412 * NO PUNCHER

No punch is connected. Program execution halts; feasibility condition: NO.

2413 * PROGRAM NOT AVAILABLE

The program present in the work area has a name different from the one specified in the order card. Program execution halts; feasibility condition: NO.

2414 * PUNCH ERROR ON CARD **

There is a punch error on the card whose immage printout is preceded by two asterisks.

2499 * END OF PUNCH CARD CHECKING

End of program.

6. EXAMPLE

The object program cards, whose name is FASE₿3 and its code is 1509, must be checked. This program is punched by •APP and inserted in the program file of the SD. The contents of the order card is the following:

Col. 1 to 4	•PCC
Col. 5 to 13	FASE₿3
Col. 14	Blank
Col. 15 to 18	1509
Col. 19 to 72	Not used
Col. 73 to 76	Optional ID code
Col. 77 to 80	Not used.

•PTR PROGRAM TRANSFER

1. FUNCTION

This program performs two different functions:

a) the generation of a type SD program file containing all or part of a basic System Disc

b) the insertion in a type SD program file of one or more programs contained in another System Disc.

The utilization of these two functions successively allows to generate a new System Disc from a basic SD and from another SD containing, in comparision with the first one, new or updated programs.

2. SYSTEM UNITS REQUIRED

This program requires the following peripherals:

- a disc unit for the System Disc used to initialize the system
- a disc unit on which output disc is mounted
- a card reader
- a printer

3. PARAMETER CARD

3.1. ORDER CARD

Col. 1 to 4	•PTR
Col. 5	must contain one of the following characters: D, if the program file of the SD must be translated fully or partially on the output disc.

I, if one or more program of the SD must be inserted in the program file on the output disc.

In the first case the order card can be followed by PRM detail cards; in the second case it must be followed by PRI detail cards.

Col. 6 to 7	SD or the logical number (col. 7) of the unit containing the System Disc.
Col. 8 to 13	serial number of the System Disc, if it must be checked; otherwise blank
Col. 14 to 15	Logical number of the unit (col. 15) containing the output disc
Col. 16 to 21	Serial number of the output disc, if it must be checked; otherwise, blank.
Col. 22 to 72	Not used
Col. 73 to 76	Optional ID code
Col. 77 to 80	Not used.

3.2. DETAIL CARDS

3.2.1. - PRM card (Program Remove) to eliminate a program

Col. 1 to 4	b̸PRM
Col. 5	type code of the program specified below
Col. 6 to 13	name of the program which must not be transferred from the System Disc to the output disc (BASIC SUB, if all subroutine library must be eliminated).
Col. 14 to 80	Not used.

3.2.2. - PRI card (Program Insert) to insert a program

Col. 1 to 4	ɃPRI
Col. 5	type code of the program specified below
Col. 6 to 13	name of the program which must be transferred from the System Disc to the output disc
Col. 14 to 80	Not used.

3.3. END OF ORDER CARD

After the detail cards must always be the end-of-order card:

Col. 1 to 4	ɃEND
Col. 5 to 80	Not used

If in col. 5 of the order card character "D" is punched and an END card follows immediately,the System Disc is duplicated on the output disc.

4. DESCRIPTION AND EXECUTION

The •PTR performs the following operations:

- check of the parameters of the order card
- check of the serial number of the input and output volumes, if requested
- search of the program file label of the SD and OD, storing useful informations and checking whether both files are type SD or not.
- analisys of the character punched in col. 5 of the order card
  - a) if the character D is present:
    - reading of the program index of the System Disc

- update and shift of the index according to the informations contained in PRM cards, if there are.
  So that the index of the programs, which must be transferred on the output disc, is generated in core store. The detail cards are read and checked formally. When the end of order card is detected, the index is written on the output disc
- if the subroutine library must not be eliminated, the index of the library is transferred on the output disc
- transfer on the OD of the program whose name is written in the index
- transfer of the subroutine, if requested.
- Update of the areas limitation on the file label of the OD

b) if the character I is present:

- reading of the PRI detail card

After checking the formal validity of the detail card, a table, containing the names of the programs which must be inserted on the OD, is built.

- If the insertion on the subroutine library is requested the index is written on the OD and the programs are shifted by a number of cylinders sufficient to contain the subroutines being inserted.

  If the available area is not sufficient the insertion of the library is not done.
- Recording of the subroutine, if requested.
- Update of the areas limitation on the file label of the OD.

5. CHECKS AND ERROR MESSAGES

The following messages may be printed:

7101 * D. UNIT N PERSISTENT ERROR

Persistent reading or writing error on disc unit N. The program halts at address 02A2. Feasibility condition: NO.

7102 * D. UNIT N OUT OF ORDER

Logical disc unit N out of order. The program causes the printer to move the paper to a new page and halts at location 01DA. Pressing START retries the operation that caused the signal. If the signal repeats, the program terminates; feasibility condition: NO.

7103 * D. UNIT N PERSISTENT ERROR

Persistent positioning error on disc unit N. Program stops at location 02A2; feasibility condition: NO.

7104 * D. UNIT N IS NOT A "SD"

No System Disc is present on disc number unit N.

7105 * 2BASICSUB MERGING IMPOSSIBLE

The insertion of the subroutine library is impossible because the available area on the output disc is not sufficient.

7110 * MULTIPLE PROGRAM NAME TNNNNNNNN

The insertion of a program already existing on the output disc is requested.

7111 * OVERFLOW OF PROGRAM FILE/TNNNNNNNN LAST PROGRAM

The free area on the output disc is not sufficient to contain all the programs being inserted. The last program inserted is TNNNNNNNN.

7114 * OVERFLOW OF PROGRAM TABLE/TNNNNNNNN LAST PROGRAM

The number of the programs contained in the output disc index has reached the maximum allowed (259).

7119 * DETAIL CARD OVERFLOW/PRI TNNNNNNNN LAST CARD

There are more than 259 PRI detail cards. The last card printed is the last card accepted by the system; these following it are ignored.

7141 * DATA CARDS WRONG

This message preceds the listing of eventual PRI detail cards not processed due to one of the following reason:

- the program to be inserted is not present in the input disc.
- the area reserved in output is not sufficient to contain the file.

7161 * DISC UNIT N NUMBER WRONG

The same logical number has been assigned to for two disc unit. Program execution stops at location 02A2; feasibility condition:NO.

7162 * UNIT N SERIAL NUMBER WRONG/SHOULD BE YYYYYY

Serial number of the volume mounted in logical unit number N is different from which specified in the order card. Program execution stops at location 02A2.

7163 * LABEL OF PROGRAM FILE NOT FOUND

The label of the program file has not been found in the output disc. Program execution stops at location 02A2.

7164 * EXECUTION TYPE CODE ERROR

An incorrect code is punched in column 5 of the order card. Program execution stops at location 02A2.

7165 * INVALID DETAIL CARD

The detail card just read has a wrong identification code (col. 1 to 4) or the program type specified is incorrect (col. 5). This message is followed by the printout of the detail card. Program execution stops at location 02A2 if the identification code is wrong, otherwise it proceedes and the card contents are ignored.

7166 * SUBROUTINES NOT FOUND

The insertion of the subroutine library has been requested but this does not exist in the input disc.

7167 * PROGRAM TNNNNNNNN NOT FOUND

The program being inserted is not present in the input disc.

7168 * PROGRAM INDEX WRONG UN. X

The program index of the disc mounted on unit X is wrong. Program execution halts at location 02A2.

7169 * SUBROUTINES INDEX WRONG

The library index in the input disc is wrong. The insertion or recording of the library is not done.

7180 * SYSTEM DISC TRANSFER

This message is printed at the beginning of the execution within one of the following specifications:

- REMOVE : if the recording of the program file with the elimination of some program is requested
- INSERT : if the insertion of one or more programs on the output disc is requested.

7198 * OUTPUT DISC INDEX

These message precedes the listing of the programs present on the output disc at end of job.

7199 * END OF SYSTEM DISC TRANSFER

End of program.

•PCK PROGRAM CHECKING

1. FUNCTION

Checking of the System Disc, obtained with the order •PTR, by means comparing new SD with basic SD.

2. SYSTEM UNITS REQUIRED

This order requires the following peripherals:

- a card reader
- a printer
- a disc unit for the basic System Disc
- a disc unit for the System Disc on which new updated program have been recorded.

3. PARAMETER CARDS

3.1. ORDER CARD

Col. 1 to 4	•PCK
Col. 5	must contain one of the following characters: D, if the program file of the SD must be transcribed fully or partialy on the output disc. I, if one or more program of the SD must be inserted in the program file on the output disc. In the first case the order card can be followed by PRM detail cards; in the second case it must be followed by PRI detail cards.

Col. 6 to 7	SD or the logical number (col. 7) of the unit containing the System Disc.
Col. 8 to 13	serial number of the System Disc, if it must be checked; otherwise, blank.
Col. 14 to 15	Logical number of the unit (col. 15) containing the output disc.
Col. 16 to 21	Serial number of the output disc, if it must be checked; otherwise, blank.
Col. 22 to 72	Not used.
Col. 73 to 76	Optional ID code.
Col. 77 to 80	Not used.

3.2. DETAIL CARDS

To check the System Disc generated by means of order •PTR, after •PCK card must be inserted the same detail cards used for •PTR order.

3.2.1. - PRM card (Program Remove) to eliminate a program

Col. 1 to 4	b̸PRM
Col. 5	type code of the program specified below.
Col. 6 to 13	Name of the program which must not be transferred from the System Disc to the output disc (BASIC SUB, if all subroutine library must be eliminated).
Col. 14 to 80	Not used.

3.2.2. - PRI card (Program Insert) to insert a program

Col. 1 to 4	ƀPRI
Col. 5	type code of the program specified below
Col. 6 to 13	name of the program which must be transferred from the System Disc to the output disc.
Col. 14 to 80	Not used.

3.3. END OF ORDER CARD

After the detail cards must always be the end-of-order card:

Col. 1 to 4	ƀEND
Col. 5 to 80	Not used

If in col. 5 of the order card character "D" is punched and an END card follows immediately, the System Disc is duplicated on the output disc.

4. DESCRIPTION AND EXECUTION

The ⊙PCK order performs the following operations:

- Checking of the parameters of the order card; the logical unit number of the output disc must not be equal to the one of the System Disc.
- Checking of the serial number of the input and output disc, if requested.
- Search of the program file label of the basic System Disc and the updated System Disc, if requested, storing useful informations.
- Storing of the index of type 1 and 3 programs present in the updated System Disc, if a 12 K core store.
- Checking of the character punched in col. 5 of the order card:

  1) if the character D is present: all detail cards, which must be PRM cards, are read and printed. After checking their validity a check is

made to see if the programs that was be eliminated still appear in the program index on the updated System Disc. Then all the program recorded on the new System Disc are compared with those in the basic SD. The subroutine library is processed as the other programs. If END card follows the order card all programs of the new SD are checked.

2) if the character I is present: a detail card, which must be a PRI card, is read and printed; the program in it specified is searched in the updated SD and compared with that in the basic SD.

5. CHECKS AND ERROR MESSAGES

The following messages may be printed:

7001 * D. UNIT N PERSISTENT ERROR

Persistent reading error on disc unit N. The program execution halts at address 02A2. Feasibility condition: NO.

7002 * D. UNIT N OUT OF ORDER

Logical disc unit N out of order. The program causes the printer to move the paper to a new page and halts at location 01DA. Pressing START retries the operation that caused the signal. If the signal repeats the program halts at location 02A2; feasibility condition: NO.

7003 * D. UNIT N PERSISTENT ERROR

Persistent positioning error on disc unit N. Program execution stops at location 02A2. Feasibility condition: NO.

7061 * DISC UNIT N NUMBER WRONG

The same logical number has been assigned for two disc unit. Program execution stops at location 02A2; feasibility condition:NO.

7062 * UNIT N SERIAL NUMBER WRONG/SHOULD BE YYYYYY

Serial number of the volume mounted on logical unit number N is different from which specified in the order card. Program execution stops at location 02A2.

7063 * LABEL OF PROGRAM FILE NOT FOUND

The label of the program file has not been found. Program execution stops at location 02A2.

7064 * EXECUTION TYPE CODE ERROR

An incorrect code is punched in column 5 of the order card. Program execution stops at location 02A2.

7065 * INVALID DETAIL CARD

Detail card wrong; the program ignores the card, which is printed, and proceeds normally.

7066 * PROGRAM TNNNNNNNN NOT ERASED

The program TNNNNNNNN has not been eliminated. This program is also checked.

7067 * PROGRAM TNNNNNNNN NOT FOUND

The Program labels are not equal; the program TNNNNNNNN is not present on one of the two discs.

7068 * INDEX WRONG

Program or subroutine index wrong. Program execution halts at location 02A2.

7069 * LENGTH OF PROGRAM TNNNNNNNN WRONG

The length of program TNNNNNNNN on the basic SD is not equal to the length of the same program on the updated SD.
The program proceeds and checks the following program.

7070 * PAGE DATA WRONG

A page of the updated SD is invalid; program proceeds normally.

7071 * PROGRAM TNNNNNNNN NOT INSERTED

The program TNNNNNNNN has not been inserted. The program proceeds normally.

7072 * SUBROUTINE YNNNNN NOT FOUND/SHOULD BE YZZZZZ

The subroutine YNNNNN is not present on the basic SD, instead the subroutine YZZZZZ is located in its place. The program proceeds comparing the subroutine index.

7073 * LENGTH OF SUBROUTINE YNNNNN WRONG

The length of the subroutine on the output disc is invalid. The program proceeds comparing subroutine index.

7080 * SYSTEM DISC CHECKING

This message is printed at the beginning of the execution whitin one of the following specifications:

REMOVE: if the recording of the program file, with the elimination of some program, is checked

INSERT: if the insertion of one or more programs is checked.

7085 * SUBROUTINE CHECKING

This message is printed at the beginning of the subroutine checking

7086 * INDEX SUBROUTINE CHECKING

This message is printed at the beginning of the index subroutine checking.

7099 * END OF SYSTEM DISC CHECKING

End of program.

•GAF SOURCE PROGRAM FILE GENERATE

1. FUNCTION

This program involves the organization and centralized operational control of a computer center's source programs in APS, COBOL and FORTRAN language, by releasing the utilization and modification of a program from the card deck making it up.

As to source programs organization, •GAF allows setting up a program library on magnetic tape, utilizing the load function of this Order program.

The file thus obtained is organized in 4005-character numbered blocks, each one of which is made up of 80 characters fixed-length logic records. Each logic record contains one source card's information, and that correspondence is shown hereafter by using the name card-record (or more simply "record").

The same block can contain card-records of different programs, and the passage from one program to another can be told from the diversity of two successive card-records' identification codes, for APS or COBOL sources, or from one particular record for FORTRAN sources (this last will be described in the section concerning FORTRAN program loading).

Regarding the operational control of an already-existing sources library, •GAF allows insertion of new programs in the file, elimination of already existing programs, updating of programs through elimination, substitution and insertion of card-records, output of programs on the printer and card punch, on the printer only, or on magnetic tape. This last function is provided to permit compiling an in-library program.
For this reason, the output tape is organized in numbered unblocked records with 85 characters. Thus the file clearly must be made up of only one program's card-records.

Utilized both for the creation and operational control of the sources library is an input tape containing the library (made up, for the first time, of header and trailer labels). Obtained in output are two tapes, one containing the created or modified library, the other containing,if required, a program extracted for compiling.

Owing to the difference between organization of FORTRAN cards in respect to APS and COBOL cards, the FORTRAN sources cannot coexist on the same tape with programs coded in COBOL and APS.

•GAF can thus organize and control either a FORTRAN library or an APS/ COBOL library.

2. SYSTEM UNITS REQUIRED

This order needs a 16K store and the following peripherals:

- a card reader
- a card punch, if required
- a printer
- a disc unit for the System Disc
- three magnetic tape units

  tape unit 4 on which the source program library is mounted

  tape unit 5 on which the updated library is to be recorded

  tape unit 1 on which the program being compiled is to be recorded, if requested.

  The last unit can be substituted by a disc unit.

3. PARAMETER CARDS

3.1. ORDER CARD

This card has diverse format, depending upon whether it requests extraction of a program or updating of the library.

Source Program File Updating

Col. 1 to 4	•GAF
Col. 5 to 7	- ƀ or any character if the Order concerns an APS/ COBOL library. - FOR code if the Order concerns a FORTRAN library.
Col. 8 to 80	Not used.

Extraction of a program existing in the library:

Col. 1 to 4	•GAF
Col. 5 to 7	- ƀ or any character if the Order concerns an APS/ COBOL program - FOR code if the Order concerns a program in FORTRAN.
Col. 8 to 15	identification code or name of the program existing in the library for which the extraction is requested.
Col. 16	code which identifies the type of extraction requested. It can assume one of the following values: - ƀ, if the program is to be recorded on tape unit 1 or on disc unit. - L, if the program is to be printed - P, if the program is to be punched and printed.

Col. 17 to 18	if col. 16 = P code which specifies punched cards' sequencing criteria. It can assume one of the following values: - b̸b̸, if the progressive numbering already existing on the library tape (numbering increment 01) is not to be changed - any number corresponding to the requested numbering increment if col. 16 ≠ P not used.
Col. 19 to 24	if col. 16 = P number to be assigned to the first punched card, leaving col. 19 blank for APS programs. b̸, if the numbering must begin from zero. if col. 16 ≠ P, not used.
Col. 25 to 30	if col. 16 = P number of the first record to be punched, leaving blank col. 25 for APS programs. b̸, if punching must begin from the first card. if col. 16 ≠ P not used.
Col. 31 to 36	if col. 16 = P number of the last record to be punched, leaving blank col. 31 for APS programs. b̸, if punching is needed up to the last card if col. 16 ≠ P not used.
Col. 37 to 80	not used.

3.2. DETAIL CARDS

These cards follow the order card only when updating the source program library.

3.2.1. - Program load card: serves to load a new program in the library.

Col. 1	Operation code: N
Col. 2 to 9	Identification code or name of the program not existing in the library and which is to be loaded on the library.
Col. 10	not used.
Col. 11 to 13	can assume one of the following values: - COB, if the program to be loaded is coded in COBOL. - ƀ, or any character if the program to be loaded is coded in APS or in FORTRAN.
Col. 14 to 80	Not used.

The card of the program to be loaded are to be placed after the Operation card. These cards must be followed by the card shown below.

Col. 1 to 5	five asterisks (*)
Col. 6 to 80	not used

Should the program to be loaded be in FORTRAN, the first source card must be the •FOR card containing in cols. 26 - 33 the name of the program specified in the N card. The name's first four characters are assigned as identification code to the card-records written on the output tape.

Each FORTRAN in-library program is preceded by a record having in positions 26 - 33 the name of the program drawn from the •FOR. This record allows distinguishing one source from another.

3.2.2. - Program elimination card: serves to eliminate a program from the library.

Col. 1	operation code: C for APS/COBOL programs R for FORTRAN programs
Col. 2 to 9	identification code or name of the program existing in the library and which is not to be transferred to the updated library tape.
Col. 10	Not used.
Col. 11 to 13	Can assume one of the following values: - COB if the program not to be transferred is coded in COBOL - ƀ or any character if the program not to be transferred is coded in APS or in FORTRAN.
Col. 14 to 80	Not used.

An elimination card,followed by a N card having in cols. 2 - 9 the identification code of the eliminated program,produces total substitution of a source library program.

3.2.3. - Program updating card: allows updating of in-library program with one or more of the following cards:

Col. 1	operation code : P
Col. 2 to 9	identification code of the program existing in the library and on which an updating operation is to be made.
Col. 10	can assume one of the following values: - 1 if the whole updated program is to be printed. - ƀ if the updated program is not to be printed.

Col. 11 to 13	can assume one of the following values: - COB if the program be updated is coded in COBOL. - ƀ or any character if the program to be updated is coded in APS or in FORTRAN.
Col. 14 to 80	Not used.

The P card must be followed by detail cards specifying the changes to be made on the program.

Given below for greater clarity are distinct descriptions of the detail cards provided, corresponding to the different languages used in the program to be updated.

a) Card record replacement card in an APS program

Col. 1	updating code: S
Col. 2 to 6	sequence number (page and line) of the record to be replaced.
Col. 7 to 72	replacement APS card
Col. 73 to 80	identification code of the program specified on the P card.

Substitution of the STRT or END card record is accepted only if the substitute card is respectively a STRT or END.

b) Elimination card of one or more card records in an APS program.

Col. 1	updating code: E
Col. 2 to 6	sequence number of the first record to be eliminated
Col. 7 to 8	ƀ
Col. 9 to 13	sequence number of the last record to be eliminated. ƀ, if only one record is to be eliminated.
Col. 12 to 72	Not used.

Col. 73 to 80	identification code of the program specified on the P card

The elimination of the STRT or END card-records is not accepted.

c) Insertion card of one or more cards between two successive card-records or after the last record of an APS program.

Col. 1	updating code: I
Col. 2 to 6	sequence number of the record after which new records are to be inserted.
Col. 7 to 72	not used
Col. 73 to 80	identification code of the program specified on the P card.

NOTE : The APS cards to be inserted, followed by an end card with an asterisk in column 1, are to be placed after the I card.

d) Replacement card of a card-record in a COBOL program.

Col. 1 to 6	sequence number (page and line) of the record to be replaced.
Col. 7 to 72	substitute COBOL card
Col. 73 to 80	identification code of the program specified in the P card.

e) Elimination card of one or more card-records in a COBOL program.

Col. 1	updating code: E
Col. 2 to 7	sequence number of the first record to be eliminated.
Col. 8 to 13	sequence number of the last record to be eliminated. ƀ if only one record is to be eliminated
Col. 14 to 72	not used
Col. 73 to 80	identification code of the program specified on the P card.

f) Insertion card of one or more cards between two successive card-records or after the last record of a COBOL program.

Col. 1	updating code: I
Col. 2 to 7	sequence number of the record after which new records are to be inserted
Col. 8 to 72	not used
Col. 73 to 80	identification code of the program specified on the P card.

The COBOL cards to be inserted having col. 1 obligatorily blank, are to be placed after the I card. The above cards must be followed by a card with an asterisk in col. 1.

g) Replacement card of a card-record in a FORTRAN program.

Col. 1 to 72	substitute FORTRAN card
Col. 73 to 76	first four characters of the name of the program specified in the P card
Col. 77 to 80	sequence number of the record to be replaced.

h) Elimination card of one or more card-records in a FORTRAN program.

Col. 1	updating code: E
Col. 2 to 5	sequence number of the first record to be eliminated.
Col. 6 to 9	sequence number of the last record to be eliminated
Col. 10 to 72	not used
Col. 73 to 80	name of the program specified on the P card.

i) Insertion card of one or more cards between two successive card-records or after the last record of a FORTRAN program.

Col.	1	updating code: I
Col.	2 to 5	sequence number of the record after which new records are to be inserted
Col.	6 to 72	not used
Col.	73 to 80	identification code of the program specified on the P card.

The FORTRAN cards to be inserted, having col. 77 obligatorily blank, are to be placed after the I card. The above cards must be followed by a card with an asterisk in col. 1 and an alphanumeric character in column 77.

General observations: The card with the asterisk in col. 1 at the end of the cards to be inserted, is not mandatory (•). If this card is missing, however, appropriate printed signaling reveals the fact.

For correct execution of program updating, the relative detail cards must be in ascending order in reference to the sequence number of the card records to be updated.

An exception to this rule is the I insertion card which, only if it follows a replacement or elimination card, can have as number of the record after which the insertion is to be made, that of the replaced or eliminated record respectively.

Regardless of the type of updating executed on a program, this is transferred to the output tape assigning to card records a new progressive sequence number.

(•) for APS and COBOL programs.

3.2.4. - Load check card: serves to check if a program was loaded correctly in the library.

Col. 1	operation code: K
Col. 2 to 9	identification code of the program existing in the library and on which the check of a previous load operation is to be made
Col. 10	Not used
Col. 11 to 13	can assume one of the following values: - COB if the program to be checked is coded in COBOL - b̸, or any character if the program to be checked is coded in APS or in FORTRAN
Col. 14 to 18	not used.

The cards of the source program being checked are to be placed after the K card. These cards must be followed by an end card with an asterisk in col. 1.

A check is made on cols. 7 - 72 for programs coded in APS and COBOL and on cols. 1 - 72 for programs coded in FORTRAN.

3.2.5. - END-OF-ORDER CARD

Col. 1	Operation code : Z
Col. 2 to 9	ZZZZZZZZ
Col. 10 to 80	not used.

This card is not necessary if the Order card refers to an extraction of an in-library program. Should the card immediately follow •GAF, duplication of the sources library is obtained on the tape mounted on unit 5.

Detail cards must be sequenced according to ascending identification code (cols. 2 - 9). Any sequencing error impedes the execution of the requested operation.

4. DESCRIPTION AND EXECUTION

The program •GAF is made of 7 segments:
The first defines the input and output files, analyses the order card, and in case of library updating, opens the files. It also contains a variety of common routines.
The second updates the APS/COBOL library.
The third updates the FORTRAN library.
The fourth retrives an APS/COBOL program.
The fifth retrives a FORTRAN program.
The sixth signals the end of reel on tape unit 5 and closes the files.
The seventh re-opens the file on unit 5, which contains updated programs, and lists them with their respective updating date. In order to list these source programs, a comment card, with the following configuration, must be inserted before the last card of each program:

a) for APS program
* (asterisk) in col. 18, followed by comment

b) for COBOL program
NOTE in col. 12 to 15, followed by comment.

c) for FORTRAN programs
C in col. 1, followed by comment.

The •GAF program proceeds in the following way:

a) if the Order card concerns the updating of the sources library, the contents of the library mounted on unit 4 are transferred to the tape mounted on unit 5, with new programs load (N card), programs elimination (C card), updating with or without total printing of the updated program (P card), check of a loading effected in a preceding job (K card).

The following operations are successively executed:

- Defining files in input and output

- Opening of the files
- Reading, checking and printing of the detail cards, and execution (if no error shown)of the function specified in each. This operation is repeated until the end-of-order card appears, which causes the transfer of any programs remaining in the library to the updated library tape.
- Eventual end-of-volume signaling for the output file.
- Closing of input and output files.
- Printing of end-of-program message.

b) If the Order card concerns extraction of a program from the library mounted on unit 5, the following operations are successively executed:

- Defining the files in input and output
- Opening of the files
- Analyzing the type of extraction requested and,if requested, of the punch routine based on the type of punch connected to the system. If no punch is connected, the impossibility of realizing the extraction requested is signalled in printing.
- Finding of the program involved in the extraction
- Eventual end-of-volume signaling, if the extraction is on tape (which,in this case,is mounted on unit 1)
- Closing of the files in input and output
- Printing of end-of-program message.

5. CHECKS AND ERROR MESSAGES

Error messages are not standard.They lack message number and HLT addresses are not definite. Message number is replaced by four asterisks.

**** MULTIPLE PROGRAM NAME

The loading of a program already existing in the library has been requested. The loading is not executed and the message for the operator is printed.

**** C/N/P/K CARD: SEQUENCE ERROR IN PROGRAM CODE

The identification code of the program specified on a detail card is out of sequence.
The requested operation is not executed and, in case of loading, updating or check, the message for the operator is printed.

**** DETAIL CARD NOT ACCEPTED: N/P/K CARD NOT FOUND

A card which is not a detail card or which is not preceded by a detail card has been read. The message for the operator is printed.

**** C/P/K CARD: PROGRAM NOT FOUND

The program specified in cols. 2 - 9 on the detail card is not present in the library. The requested operation is not executed and, in case of updating or check, the message for the operator is printed.

**** N/P/K CARD NOT ACCEPTED: DETAIL CARD NOT FOUND
A loading, updating or check card not followed by a detail card has been read.

**** PUNCHING IMPOSSIBLE

The extraction of a program on the punch while no punch is connected. to the systems, has been requested.

**** CHECK OF NNNNNNNN PROGRAM O.K.

The check of the NNNNNNNN program was successful.

****CHECK OF NNNNNNNN PROGRAM FAILED

In this case, before signalling, the tape and records causing the error are printed, respectively preceded by the words RECORD and CARD. Any excess of records over cards or of cards over records shows up when one of the two printed records is made up of blanks.

****E/I/S CARD: PROGRAM CODE DOES NOT AGREE WITH "P" CARD

The identification code punched on an E/I/S card does not correspond to the code of the P card preceding it. The request is not executed.

**** SEQUENCE ERROR IN E/I/S CARD

The sequence number pertaining to updating is out of sequence. The requested operation is not executed.

**** E/I/S CARD: RECORD NOT FOUND **** LAST RECORD ****

The sequence number of the records to be updated finds no corresponding number in the records of the in-library program.
The sequence number of the last record found is signalled. The request is not executed and the message for the operator is printed.

**** ASTERISK CARD NOT FOUND

The cards following an insertion card do not have an end card with an asterisk in col. 1

**** REPLACED RECORD

This message precedes the printing of each record replaced in a program.

**** REPLACEMENT

This signaling precedes the printing of each record replacing.

**** REMOVED

This signaling precedes the printing of each record eliminated from a program.

**** MERGING

This message precedes the list of records inserted between two successive records or after the last record of a program.

**** REMOVE DETAIL CARDS UP TO THE FIRST C/N/P/K/Z CARD

Message for the operator so that he may extract from the deck of unread cards those preceding the first detail card not yet considered. The program stops on a wait HLT. By depressing the START button, the operator can in any case arrange for the program to ignore the cards to be extracted until a new detail card appears.

**** FOR CARD NOT FOUND

The loading of a FORTRAN program not preceded by the •FOR card is requested.
The loading is not executed and the message for the operator is printed.

**** WRONG E/S CARD

The elimination of the STRT or END card-record or else its substitution with a card different from STRT or END respectively, is requested.
The request is not executed. This message concerns only programs coded in APS.

**** END OF VOLUME ON UN.X

The tape mounted on the unit X is not sufficient.

**** PROGRAM NOT FOUND

The program to be extracted is not found on the tape mounted on unit 5.

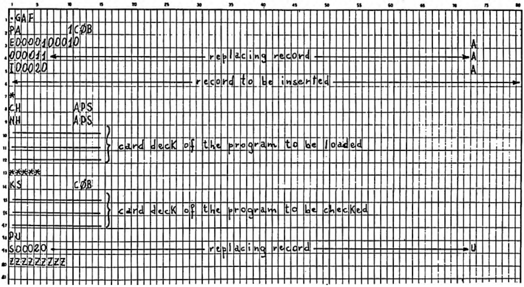
**** END G.A.F. *

End of program.

a) An APS/COBOL source program library is to be updated with the following operations: check of the program S, substitution of the program H, updating with printing of the program A with elimination. of the first ten records, substitution of the eleventh, insertion of a record after the twentieth, and updating of the program U with substitution of the twentieth record.

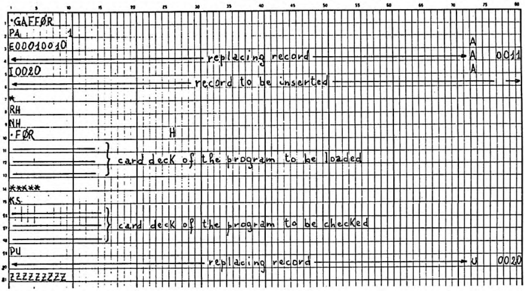
The program A and S are coded in COBOL, while the programs H and U are coded in APS.

The following cards must be punched:



```
-GAF
PA       1COB
E0000100010                                                             A
000011 <-------------------- replacing record -------------------->     A
I00020                                                                  A
<-------------------------- record to be inserted --------------------------->
*
CH        APS
NH        APS
______________ }
______________ } card deck of the program to be loaded
______________ }
*****
KS        COB
______________ }
______________ } card deck of the program to be checked
______________ }
PU
S00020 <-------------------- replacing record -------------------->     U
ZZZZZZZZZ
```

b) A FORTRAN source program library is to be updated by executing the same operations described in the preceding problem. The following cards must be punched.



```
 1 ·GAFFØR
 2 PA       1
 3 E00010010                                                               A
 4 <---------------------------- replacing record ------------------------> A   0011
 5 I0020                                                                   A
 6 <-------------------------- record to be inserted ------------------------------>
 7 *
 8 RH
 9 NH
10 ·FØR                     H
11 ---------------  }
12 ---------------  } card deck of the program to be loaded
13 ---------------  }
14 *****
15 KS
16 ---------------  }
17 ---------------  } card deck of the program to be checked
18 ---------------  }
19 PU
20 <---------------------------- replacing record ------------------------> U   0020
21 ZZZZZZZZZ
```

c) Extraction on tape of program COPIA codified in FORTRAN language.

Col. 1 to 4	●GAF
Col. 5 to 7	FOR
Col. 8 to 15	COPIAƀƀƀ
Col. 16 to 80	blank

d) Extraction, on punch and printer of program COPIA, codified in APS or in COBOL, with numbering increment 5

Col. 1 to 4	•GAF
Col. 5 to 7	blank
Col. 8 to 15	COPIAƀƀƀ
Col. 16	P
Col. 17 to 18	05
Col. 19 to 80	blank

•LPC LOAD PROGRAM BY CARDS

1. FUNCTION

The Load Program by Cards loads and starts a program on punched cards. The object program must be the same in make up as required for an •SDU program, or generated as a •APP output program.

The program to be started, of type 1 or 3, can include several segments to be read in the logical sequence in which they are called.

2. SYSTEM UNITS REQUIRED

Load Program by Cards requires the following peripherals:

- a disc unit receiving a SD, LD, or MD
- a card reader
- a printer.

3. PARAMETER CARDS

Load Program by Cards requires a •LPC order card and two types of detail cards in addition to the LBC cards if required and the END card; a PLC card when the object program is initiated and an SIN card at the beginning of each segment to be initiated. The cards must be arranged in the following order:

- •LPC order card
- LBC opening cards if the program to be initiated is to use files on disc or tape (these cards must then be followed by an END card).

- PLC card
- cards of the object program to be loaded; a SIN card must begin each segment
- END of order card.

3.1. ORDER CARD

It has the following format:

Col. 1 - 4	identification code: ●LPC
Col. 5 - 12	program name
Col. 13 - 14	not used
Col. 15	- H when a HLT preceding the execution of the program is desired - blank otherwise
Col. 16	- b̸ if the store is to be filled with characters 0A before the program is loaded - * (asterisk) otherwise
Col. 17	- * if the storing in User's Area of data contained in columns 18 - 49 is desired - b̸ otherwise
Col. 18 - 49	user data, (if any)
Col. 50 - 72	not used
Col. 73 - 80	serial code, (if any).

3.2. DETAIL CARDS

The program requires the PLC and SIN detail cards and the END of order card.

3.2.1. PLC detail Card (Program Label Create)

This card carries out the same functions as the program label. It has the following format:

Col. 1 - 4	identification code: b̸ PLC
Col. 5	program type: - 1 for DOS program - 3 for user program
Col. 6 - 13	program name
Col. 14	indicates the type of input/output routines used by the program. Has the following values: b̸ - magnetic tape management routine 1 - disc management routine 2 - both magnetic tape and disc management routines 3 - neither magnetic tape nor disc management routine
Col. 15 - 23	Not used
Col. 24 - 27	file table address for the program (coded hexadecimal) 0000 - if no file table
Col. 28 - 31	program start address (coded hexadecimal)
Col. 32 - 35	maximum program address + 1 (coded hexadecimal)
Col. 36	I code
Col. 37 - 80	not used.

3.2.2. SIN Detail card (Segment insert)

As it has been said before there must be a SIN card preceding each segment.

The card has the following format:

Col. 1 - 4	identification code: ƀSIN
Col. 5	program type: - 1 for DOS program - 3 for user program
Col. 6 - 13	program name
Col. 14 - 19	not used
Col. 20 - 21	number of the segment referred to the object program cards that follow theSIN card; if it is the first segment 01 is punched in this field.
Col. 22 - 79	not used
Col. 80	character E for end of segment

3.2.3. End of Order Card

The last card in the deck must be the END card. It has the following format:

Col. 1 - 4	identification code: ƀEND
Col. 5 - 79	not used if the program to be initiated is a type 3 program (i. e. user program) if it is a type 1 program (i. e. DOS program) punch parameters as for order card.
Col. 80	character F for end of program.

4. DESCRIPTION AND EXECUTION

The •LPC program, made up of two segments is recorded on cylinder zero of the volume used as the system source disc.

The first segment is initiated by Pilot 1 and is located at address 4608 (decimal).

The second segment is read by the first one and is stored at address 5400 (decimal).

The first segment performs the following functions:

- reads and stores the LBC cards, if any
- assigns logical numbers to any tape handlers, connected to the system
- stores into memory the start address of the disc area which contains the general checkpoint file or the special file, if present. Otherwise five binary zeros are recorded
- checks the size of the disc area containing the CHKP file: it must not be less than twice the size of the memory area
- stores into memory the segment load routine from cards, starting at address 684 (decimal)
- stores into memory the PLC card and the SIN card relative to the first segment
- when the program to be initiated involves files on disc/tape, it reads into memory at the program location address as many object program cards as are necessary to make up the file table
- stores and initiates the second •LPC program segment.

The second segment performs the following functions:

- processing of LBC cards, if any stored in memory (See paragraph A5 in the Appendix).

- validation of a HLT instruction prior to the initiation of the program when code H punched in column 15 of the order card (HLT is stored at the address 01D2)

- storing into the memory print area the last part of the •LPC program which normalizes the memory with the characters 0A, if required by code b̸ (blank) punched in column 16 of the order card

- calling into execution of the segment load routine from cards.

The segment load routine from cards reads memory the remaining cards of the first segment of the program and initiates it.

5. CHECKS AND ERROR MESSAGES

After checks have been performed on the working of the disc unit and the validity of the order cards, errors are indicated by means of the following messages:

2001 x	Persistent error in reading on the disc unit containing the system volume. Program stops; feasibility condition: NO.
2002 x	The disc unit containing the system source volume signals an "out of order" condition. Program makes a page change on printer and halts at the address 01DA. After RUN has been depressed the program repeats the operation which caused the message; if the message reappears, the program stops. The feasibility condition is: NO.
2003 x	Persistent error in positioning on the disc unit containing the system source volume. Program stops; feasibility condition: NO.

2010 x	Option incorrectly written in •LPC card. Program stops; feasibility condition: NO.
2011 x	Detail card with incorrect identification code. Program stops; feasibility condition: NO.
2012 x	Program name on a detail card does not match the name in the other cards. Program stops; feasibility condition: NO.
2013 x	Columns 20 - 21 of the first card SIN contain a segment number other than 01. Program stops; feasibility condition: NO.
2014 x	The size of the area on which the CHECKPOINT file is recorded is insufficient. This message is also used when the program does not include use of such a file. The program is loaded and initiated. Feasibility condition:YES.
2015 x	Format error detected in processing on LBC card; program stops. Feasibility condition: NO.
2016 x	An overflow condition is found in the LBC card storage area. This area is big enough to contain LBC cards to open as many files as can be processed by a program, that is 10. Program stops; feasibility condition: NO.
2099 x	START OF PROGRAM XXXXXXXX This message indicate that control is transferred to the card segment load routine which completes loading of the first segment and initiates it.

6. EXAMPLE

It is desired to initiate a type 3 program called LIST1 punched on cards. This program is constituted by made up a single segment that uses card read and print routines only. The program start and end addresses are 2048 and 4095 respectively (both decimal) and the number of cards is 100. In addition it is desired to normalize the memory with characters 0A before the program is read. Program initiation is to be preceded by a HLT.

The following cards must be punched:

1. Order Card

Col. 1 - 4	●LPC
Col. 5 - 12	LIST1
Col. 13 - 14	not used
Col. 15	H
Col. 16 - 80	not used.

2. PLC Detail Card

Col. 1 - 5	ƀPLC
Col. 5	3
Col. 6 - 13	LIST1
Col. 14	3
Col. 15 - 23	not used
Col. 24 - 27	0000
Col. 28 - 31	0800
Col. 32 - 35	1000
Col. 36	I
Col. 37 - 80	Not used.

3. Detail Card SIN

Col. 1 - 4	ƀSIN
Col. 5	3
Col. 6 - 10	LIST1
Col. 11 - 15	not used
Col. 16 - 19	0100
Col. 20 - 21	01
Col. 23 - 79	not used
Col. 80	E

4. END card

Col. 1 - 4	ƀEND
Col. 5 - 79	not used
Col. 80	F

The cards must be read in the order indicated above. The object program cards must be inserted between the SIN card and the END card.

•DPG DEBUGGING PROGRAM GENERATE

1. FUNCTION

To generate, from an existing type 3 program of the system disc, a program for controlled execution. Such generation includes insertion of corrections, in basic language, and the trace routine, if requested, plus the various-re-entry points requested by the user, in addition.

The type 3 program selected by the user is taken from the program file. When the controlled program is generated it is written at the end of the program file, i. e. after the other programs that have already been processed by the order •DPG; the original program is left unchanged. These programs are shown as type C programs. A type C program cannot be further treated by the order •DPG. To insert additional corrections in a type C program, the original version must always be used. If an earlier type C program exists on the System Disc, the new version is inserted at the end of the file but does not delete the previous one. In this case, the earlier version can be deleted by an order •SDS.

If a user should so request, an order •DPG can delete all type C programs on the System Disc.

Type C program cannot be transferred to the Library Disc or to the Master Disc: its function is limited to the debugging phase is then started by an order •PRL (Program Load), with a C punched in column 13 of the order card.

In requesting a certain number of corrections and use of the trace routine with its various re-entry points, the user must bear in mind that insertion of such requests depends on the memory area left available by the program and that, in any case, the •DPG program can only insert a maximum of 15 trace points.

The trace routine must take up a memory area outside that of the program and be located at an address that is a multiple of 256 with a length of 468 characters. An index, whose length is variable and depends upon the number of re-entry points requested, is inserted at the end of this area. (This index, as will be seen, is made up of n records each containing 12 characters).

It should be noted that for each re-entry point requested by the user, a statement 4 or 6 characters in length must be provided. This statement can be replaced by a return jump to the trace routine.

It should also be remembered that a return jump to the trace routine destroys the contents of memory locations 254 and 255 which previously existed.

2. SYSTEM UNITS REQUIRED

The •DPG program requires the following peripherals:

- a disc unit on which the SD is located
- a printer
- a card reader to read detail cards.

3. PARAMETER CARDS

3.1. ORDER CARD

The format of the order card is:

Col.	1 to 4	Order identification code: •DPG
Col.	5 to 12	Name of the type 3 program to be generated for controlled execution.

Col. 16	x - Delete all controller execution programs existing in the SD. b̸ - No deletion.
Col. 14 to 72	Not used.
Col. 73 to 80	Serial code (if desired).

3.2. DETAIL CARDS

There are three types of detail cards, depending on whether only corrections are to be inserted (CRT), or the trace routine is to be inserted.

3.2.1. CRT Card (Correction)

The format of the CRT cars is as follows:

Col. 1 to 4	Identification code: b̸CRT
Col. 5 to 6	Segment number to which the correction refers (value in decimals).
Col. 7 to 8	Correction length, in coded hexadecimal form
Col. 13 to 72	Correction in base language, in coded hexadecimal form
Col. 73 to 80	Not used.

3.2.2. DTR Card (Trace routine)

The DTR card is used to insert the trace routine.

The format of this card is as follows:

Col. 1 to 4	Identification code: b̸DTR
Col. 5 to 8	Location address of the trace routine, in coded hexadecimal form.

This address must be a multiple of 256.

Col. 9 to 80 Not used.

3.2.3. TRC Card (trace routine call)

The DTR card requests insertion of the type C routine of the trace routine, whereas, the TRC card indicates where the tracing is to occur and what memory area to print. One TRC card is therefore needed for each trace routine entry point.

The TRC card has the following format:

Col. 1 to 4	Identification code: ɓTRC
Col. 5 to 6	Segment number in which the jump is located.
Col. 7 to 10	Memory Address of the jump point location, in coded hexidecimal form.
Col. 11 to 14	Initial address of the memory area to be printed in coded hexidecimal form.
Col. 15 to 18	Length of the memory area to be printed, in coded hexidecimal form.
Col. 19 to 80	Not used.

Detail cards are printed during reading; they should be placed in the reader in the following order:

a) CRT cards in ascending segment number order

b) DTR card, if any

c) TRC cards, if any, in ascending segment number order

d) END card.

4. DESCRIPTION AND EXECUTION

The •DPG program is made up of three segments at memory location 2048 (in decimal form).

It first scans the program file directory for the program called by the order card. To do so, it reads column 13 for the SD location address of the generated program. Character corrections, or instructions which are already part of the segment, or additional corrections are inserted with the possibility of being grouped into blocks, at the end of the corresponding segment. Then, if requested, the trace routine and the corresponding jumps are inserted. After the trace routine has compiled addresses indicated in the DTR card, the routine is located, with whatever corrections may be applicable, at the end of the first segment, in a 468 character page. Jump returns are inserted in the program as though they were corrections, at the entry points indicated on cards TRC, plus the •DPG control program; so that the latter are as on the CRT cards, they are arranged in segment number order.
At the end of the first segment, therefore, i.e. immediately following the trace routine, a jump index is made up of the same number of 12-character records (maximum of 15) as there are entry points in the program. A record in the index is made up as follows:

Card 1 to 2	Input address of the trace routine,
Cards 3 to 4	Initial address of the memory area to be printed,
Cards 4 to 5	End address of the area to be printed,
Cards 6 to 11	Program instruction including the return jump location of the trace routine.

While a new program is being inserted, the •DPG checks that the file is not greater thant the area reserved for it; the label is updated by allowing room for corrections, the trace routine and the index, to determine the new theoretical memory location of the program.

5. CHARACTERISTICS OF THE TRACE ROUTINE

Once this routine has been called by the return jump inserted by the •DPG program, it scans the jump index and picks out data concerning the memory area to be printed. If this scan is without result, such as in the case of a modifation of the index by the program, the trace routine stops at a HLT and, when RUN is pressed, sets the feasibility condition to NO. The trace routine usually prints the print area first, and it is modified after wards. It then prints a line containing the address of the program point as follows:

TRC I I I I   where I I I I is the binary value in hexadecimal form.

Next the memory is printed in the same form as that of the •MDP program. At the end, the trace routine returns registers UF/OF and ZE/NZ to their initial status, carries out the program instruction which was replaced by a jump return and returns to the main program.

6. CHECKS AND MESSAGES

The following messages are transmitted when the peripherals are checked and the contents of the parameter cards read:

1101 x D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or verifying logical disc unit N. Program terminates; feasibility condition: NO.

1102 x D. UNIT N OUT OF ORDER

Logical disc N out of order.
The program causes the printer to move the paper to a new page and the program halts at location 01DA. Pressing RUN retries the operation that caused the signal.
If the signal repeats, the program terminates; feasibility condition: NO.

1110 x PROGRAM NOT FOUND

SD Program file directory does not contain a program with the name punched in the •DPG card.
Program terminates; feasibility condition: NO.

1111 x XXX CARD ERROR

Type XXX detail card wrong.
Program terminates; feasibility condition: NO.

1112 x XXX CARD SEQUENCE ERROR

Type XXX detail card not in the proper order (e.g. correction cards not arranged in ascending segment number order).
Program terminates; feasibility condition: NO.

1113 x SUB-ORDER CARD ERROR

Card read cannot be identified.
Program terminates; feasibility condition NO.

1114 x TRC-POINT ERROR

An instruction having a length of 2 is found at the specified program address as given in the detail card TRC.
Program terminates; feasibility condition: NO.

1115 x OVERFLOW OF TRC TABLE

More than 15 entry points requested on the trace routine.
Program terminates; feasibility condition: NO.

1116 x OVERFLOW OF GENERATE PROGRAM

The generated program exceeds the size of the memory.
Program terminates; feasibility condition: NO.

1117 x SEGMENT NOT FOUND

Program does not contain a segment with the same number as indicated in the TRC or CRT cards.
Program terminates; feasibility condition: NO.

1118 x OVERFLOW OF PROGRAM

The program file, after the generated program is inserted, exceeds the area reserved for it. Program terminates; feasibility condition: NO.

1198 x PROGRAM FILE NOT AVAILABLE

This message indicates that due to the last error indication, the program file can no longer be used.

7. EXAMPLE

To carry out a basic language correction to the program FASE2 and delete all debugging programs from the file, and replace two 6-character instructions by two other similar ones, the order card must contain the following:

Col. 1 to 4	●DPG
Col. 5 to 9	FASE2
Col. 10 to 12	b̸ . . . b̸
Col. 13	ẋ
Col. 14 to 80	b̸ . . . b̸

and must be followed by a detail card CRT with the following punches on it:

Col. 1 to 4	b̸CRT
Col. 5 to 6	01

Col. 7 to 8	0C
Col. 9 to 12	0804
Col. 13 to 36	D201 07B4 07FE D201 07F6 0795
Col. 37 to 80	b̸ . . . b̸

The END card must be punched as follows:

Col. 1 to 4	b̸END
Col. 5 to 80	Blank.

⊙DIL DISC LOAD

1. FUNCTION

To write data on disc from card input; for data on a program file, cards will have a "program" format, and for data on a data file, cards will have a "data" format.

2. SYSTEM UNITS REQUIRED

The following peripherals are needed:

- a disc unit for recording data,
- a disc unit for the System Disc, if the unit to be used for recording is not the System Disc,
- card reader,
- printer.

3. PARAMETER CARDS

3.1. ORDER CARD

The format of the order card is:

Col. 1 to 4	Order identification code: ⊙DIL
Col. 5 to 6	Logical disc unit (punched in column 6) containing the volume to be recorded; punch SD if the volume is the System Disc.
Col. 7	b̸ if the recorded cards are not to be printed. 1 or any other character to print cards.

Col. 8 to 72	Not used
Col. 73 to 80	Serial code (if desired).

3.2. DETAIL CARDS

Cards for recording are arranged in pages with a maximum length of 2048 characters; at the end of the card deck, an END card indicate the end of the job.

3.2.1. "Program" Cards

Each group of cards making up one disc page must be preceded by a "beginning of page" card, having the following format:

Col. 1 to 5	Disc address (in decimal form) of the page beginning. Right hand zeros must be punched as well.
Col. 6	Write recording mode - 1 for 8-bit mode - 2 for 6-bit mode.
Col. 80	*

The cards making up disc page have the same format as packed APS cards (in pure or coded hexidecimal form).

Col. 1 to 4	Code of the program to be loaded (this area is not used by •DIL).
Col. 5 to 6	Length (in hexadecimal) in memory characters of the part of the program punched in columns 11 to 70.
Col. 7 to 10	Address of memory insertion of the part of the program punched in columns 11 to 70.

Col. 11 to 70	Part of the program to be recorded
Col. 71 to 79	Not used
Col. 80	b̸ or any other character except S and *.

3.2.2. "Data" Cards

Each group of cards making up a page is preced by a "beginning of page" card. This card is easily recognizable, as it contain punches in the first 8 columns, whereas all other cards for that page are blank in the first 8 columns.

All "data" cards have an S in column 80.

The format for data cards is:

Col. 1 to 5	Sector number at which the page begins. All right hand zeros must be punched.
Col. 6	Recording mode: - 1 for 8-bit mode - 2 for 6-bit mode
Col. 7 to 8	Number of cards making up page including beginning of page card. All right hand zeros must be punched as well. Data included in columns 1 through 8 are only to be punched in the first card of each page.
Col. 9 to 10	Length, in memory characters, of data punched in columns 11 to 70. Right hand zeros must also be punched.
Col. 11 to 70	Data to be written
Col. 71 to 78	Not used
Col. 79	Mode of punching. - 1 for hexadecimal - 2 for alphanumeric.
Col. 80	S

4. DESCRIPTION AND EXECUTION

This program is made up of a single segment located at address 2048 (decimal); the program functions in two different ways, depending on whether a program or data file is to be written.

Data is always transcoded (if it is not in pure hexadecimal) and unpacked.

Once a "beginning of page" card is read and the information it contains is recorded, the first data card is read and the storage address punched in the card is recorded. The data is then entered into the memory at the beginning of the page area and over the length indicated. The following cards are then located on the page at their relative addresses, using the address of the first card and the address punched in the card as the bases. Extra locations remaining after a page is recorded are filled with spaces.

The program checks that no page has cards with an input address lower than that of the beginning page card (message N° 0513) and that no page exceeds the maximum length of 2048 characters (message N° 0514).

A page formulated in the memory in this way is recorded on a disc for every card containing an asterisk (*) in column 80 or for every END card read.

4.2. "DATA" CARDS

If column 79 contains a 1, data is converted (if it is in coded hexadecimal) and packed; if column 79 contains a 2, the data is stored in the page as punched.

Cards are entered in a page one after the other; the page is recorded on a disc whenever the number of cards read is equal to the number punched in columns 7 and 8 of the beginning of page card. Once a page is recorded, reading of a card which is not a beginning of page card or an END card (message No 0510) is signalled as an error; error signals are also caused whenever a card with a punch other than S in column 80 is read before the proper card number is passed, or if a card indicating the beginning of a new page is read (message N° 0516).

5. CHECKS AND ERROR MESSAGES

Program checks on peripheral unit operation, on order card or detail card validity, and may give rise to the following printed messages:

0501 * D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or verifying logical disc n. Program terminates; feasibility condition: NO.

0502 * D. UNIT N OUT OF ORDER

Logical disc unit n signalled out of order. The program causes the printer to move the paper to a new page and the program halts at location 01DA. Pressing RUN retries the operation that caused the signal. If the signal repeats, the program terminates. Feasibility condition: NO.

0503 * D. UNIT N PERSISTENT ERROR

Positioning error on logical disc unit n. Program terminates; feasibility condition: NO.

0510 * FIRST BLOCK CARD WRONG

Indicates an error on beginning of page card (for "program" or "data" file). This message is caused by one of the following errors:

a) Error in column 80.

b) Columns 1 to 10 of a beginning of page card containing an S in column 80 have punches that are not numerical or blank spaces.

c) Disc Address (Col. 1 to 5) greater than 19999.

d) Length of a data greater than 60 or equal to 0 on cards with an S in column 80.

e) Column 1 to 6 of a card with an asterisk (*) in column 80 contain non numeric characters.

f) Column 6 contains punches other than 1 or 2.

The program terminates; feasibility condition: NO.

0511 x COLUMN 80 WRONG

Program card (other than a beginning of page card) has an S or x in column 80.
Program terminates; feasibility condition: NO.

0512 x OBJ. PROG. CARD NOT EXAD

Program card (other beginning of page card) has punches that are not hexadecimal .
Program terminates; feasibility condition: NO.

0513 x OBJ. PROG. CARD ADD. TO LOW

Program card has an address inferior to that of the page. Program terminates; feasibility condition: NO.

0514 x OVERFLOW OF BLOCK

The last character to be stored exceeds the maximum page size (2048 characters). Program terminates; feasibility condition: NO.

0515 x OBJ. PROG. CARD LENGTH WRONG

The area punched in columns 5 and 6 of a program card, which represent the length of the data, is equal to zero or greater than 30.
Program terminates; feasibility condition: NO.

0516 x DATA CARD WRONG

This error message may be caused by one of the following cases:

a) A beginning of card page is read before all the data cards for a page have been read.

b) The area punched in columns 9 and 10 of a data card contains non numeric values.

c) The value of the area punched in columns 9 and 10 of a data card is equal to zero or greater than 60.

Program terminates; feasibility condition: NO.

0517 x COL. 79 DATA CARD WRONG

Col. 79 of a data card does not contain a 1 or a 2.
Program terminates; feasibility condition: NO.

0518 x DATA CARD NOT EXAD

Non hexadecimal characters are punched in the data area (columns 11 to 70) of a data card with a 1 in column 79.
Program terminates; feasibility condition: NO.

0519 x ORDER CARD WRONG

Error in the format of an order card.
Program terminates; feasibility condition: NO.

6. EXAMPLE

To record a sample file on the System Disc and then to print cards making up this file, the order card should have the following structure:

Col. 1 to 4	●DIL
Col. 5 to 6	SD
Col. 7	5

This will be followed by the sample file cards, punched in the same way as "data" cards, and then by the END card.

•DIP DISC PRINT

1. FUNCTION

To print specified disc areas containing any kind of data.

2. SYSTEM UNITS REQUIRED

The Disc Print program uses the following peripherals:

- a disc unit containing the areas to be printed
- a disc unit containing the System Disc, if the data to be printed is not recorded on SD
- a printed.

3. PARAMETER CARDS

The Disc Print program uses only the order card with the following format:

Col. 1 to 4	Order identification code: •DIP
Col. 5 to 6	Logical disc unit (punched in column 6) of the unit containing the volume to be printed; if this is the System Disc, punch SD.
Col. 7	Recording mode: - 1 for 8-bit mode - 2 for 6-bit mode

Col. 8 to 12	Initial sector of the area to be printed. Punch all right hand zeros as well.
Col. 13 to 17	Final sector of the area to be printed. Punch all right hand zeros as well.
Col. 18 to 28	Five other possible parameter groups for printing other areas, when required.
Col. 29 to 39	
Col. 40 to 50	
Col. 51 to 61	
Col. 62 to 72	
Col. 73 to 80	Serial code (if desired).

4. DESCRIPTION AND EXECUTION

This program is made up of a single segment located at address 2048 (decimal).

Printing starts with the volume label; then come the various areas, sector by sector, as requested and with the same structure as the "memory print" order. The corresponding address is printed in the first five columns of the first line.

As can be deduced by their format, order cards can print up to a maximum of six areas each.

5. CHECKS AND ERROR MESSAGES

Checks on peripheral units and on card punches may give rise to the following messages:

0401 x D. UNIT N PERSISTENT ERROR

Persistent error in reading logical disc unit N.
Program terminates; feasibility condition: NO.

0402 * D. UNIT N OUT OF ORDER

Logical disc unit N signalled out of order.
The program causes the printer to move the paper to a new page and halts at location 01DA.
Pressing RUN retries the operation that caused the signal.
If the signal repeats, the program terminates and the feasibility condition is: NO.

0403 * D. UNIT N PERSISTENT ERROR

Persistent positioning error on logical disc unit N.
Program stops; feasibility condition: NO.

0410 * ORDER CARD WRONG

Wrong parameters in columns 7 to 72.
Program stop; feasibility condition: NO.

0411 * DISC UNIT N NUMBER WRONG

Wrong parameter in columns 5 and 6 of the order card.
Program stops; feasibility condition: NO.

6. EXAMPLE

To print the following areas of the SD:

- sectors 00002 to 00019
- sectors 19630 to 19642 in 8-bit mode
- sectors 19690 to 19791 in 8-bit mode

The order card should be punched as follows:

Col. 1 - 4	●DIP
Col. 5 - 6	SD
Col. 7	2
Col. 8 - 12	00002
Col. 13 - 17	00019
Col. 18	1
Col. 19 - 23	19630
Col. 24 - 28	19642
Col. 29	1
Col. 30 - 34	19690
Col. 35 - 39	19791
Col. 40 - 72	b̸ . . . b̸
Col. 73 - 80	b̸ . . . b̸

•FLP FILE LABEL PRINT

1. FUNCTION

To print the labels for files within a volume, the volume label and the free area table.

2. SYSTEM UNITS REQUIRED

The File Label Print program requires the following peripherals;

- a disc unit on which the System Disc is installed
- another disc unit, if labels to be printed are not on the SD
- a printer.

3. PARAMETER CARDS

The program requires only the order card with the following format:

Col. 1 to 4	•FLP
Col. 5 to 6	Logical disc unit number (punched in column 6) of the unit on which the volume containing the labels to be printed is installed: SD if this is the System Disc
Col. 7 to 72	Not used.
Col. 73 to 80	Serial code (if desired).

4. DESCRIPTION AND EXECUTION

The File Label Print program is made up of a single segment stored at location 2048 (decimal) in the memory.

5. CHECKS AND ERROR MESSAGES

The program checks peripheral unit operation and the validity of the parameter card and, when an error is found, prints one of the following messages:

1201 x D. UNIT N PERSISTENT ERROR

Persistent error in reading logical disc unit N. Program terminates; feasibility condition: NO.

1202 x D. UNIT N OUT OF ORDER

Logical disc unit N signalled out of order.
The program causes the printer to move the paper to a new page and halts at location 01DA.
Pressing RUN retries the operation that caused the signal. If the signal repeats, the program terminates; feasibility condition: NO.

1203 x D. UNIT N PERSISTENT ERROR

Positioning error on logical disc unit N. Program terminates; feasibility condition: NO.

1210 x ORDER CARD ERROR

Columns 5 & 6 of the parameter card contain wrong punches. Program terminates; feasibility condition: NO.

1211 x NOT FOUND UNIT N

No logical disc unit N included in peripherals. Program terminates. feasibility condition: NO.

6. EXAMPLE

To print the labels of the volume installed on logical, disc unit 1, the order card should be punched as follows:

Col.	1 to 4	•FLP
Col.	5 to 6	ƀ1
Col.	7 to 80	Blank.

•CRD CARD DUMP

1. FUNCTION

To print all cards following the order card until an end of print card appears.

2. SYSTEM UNITS REQUIRED

The Card Dump program uses the following peripherals:

- a disc unit for the SD
- a printer
- a card reader.

3. PARAMETER CARD

The Card Dump Program uses an order card and a detail card.

3.1. ORDER CARD

Col.	1 to 4	Order identification code: •CRD
Col.	5 to 72	Not used
Col.	73 to 80	serial code.

3.2. END OF PRINT CARD

This is the last card in the deck of cards to be printed.

Col. 1 to 4	Identification code: b̸...
Col. 5 to 80	Not used.

4. DESCRIPTION AND EXECUTION

The program reads and prints at the same time in the following manner: a card is read, stored in the print area and, when a new card is being read, the preceding card is printed. This process continues until the end of print card is read.

All cards are printed, including the order card, and a line space is left between each card.

5. CHECKS AND ERROR MESSAGES

To read cards, the program uses the card read and print routine. In case of a print error, the character ≠ is printed in the first position of the line containing the error. The following message is printed at the end of the job:

```
2299 * END OF CARDS DUMP
```

6. EXAMPLE

To print a given deck of cards, an order card is punched and placed at the beginning of the deck and an end of print card is placed at the end. These cards are punched as follows:

1. Order Card

Col. 1 to 4	•CRD
Col. 5 to 80	Blank

2. **End of Print card**

Col.	1 to 4	b̸ . . .
Col.	5 to 80	Blank.

•MDP MEMORY DUMP

1. FUNCTION

To print the contents of a certain specified area of the memory as requested by the user. The whole memory can be printed, or the reserved area or the area available for programs.The reserved area is regenerated each time Pilot Zero is loaded, therefore the list of the reserved area has no meaning.

In order to print the reserved area must be used the COS memory dump program (see Utility Programs/COS; program code 0201).

To print the part of the reserved area preceding 01C0 must be used the COS memory dump program (see Utility Programs/COS; prog.code 0219).

2. PARAMETER CARDS

Memory Dump requires only the order card with the following format:

Col. 1 to 4	Order identification code •MDP
Col. 5 to 8	Initial address of the memory area to be printed; address in coded hexadecimal.
Col. 9 to 12	Final address
Col. 13 to 72	Not used.
Col. 73 to 80	Serial code (if desired).

If the initial address is not specified, printing begins with the first address left free by the programs; if the final address is not specified, printing stops at the end of the memory.

3. DESCRIPTION AND EXECUTION

The order is executed by the memory print program which is part of the Basic Program and is initiated by Pilot 1.

Memory contents are printed out in two lines: one hexadecimal and the other in graphic characters in their normal form.

First the name of the last program being executed is printed out in the following manner:

```
9012 * MEMORY DUMP OF PROGRAM X . . . X
```

At the end of printing, the following message is printed:

```
9099 *END OF MEMORY DUMP.
```

●SPD SYSTEM DISC PRINT AND DUPLICATE

1. FUNCTION

To duplicate, on the volume indicated, the program file located on the System Disc or the Library Disc and if required, to print a program file index and the index of the library of Subroutines forming part of the file, or simply to print the index without duplication.

2. SYSTEM UNITS REQUIRED

The System Disc Print and Duplicate program uses the following peripherals:

- a disc unit on which the SD or LD is mounted
- a second disc unit if duplication is desired
- a printer.

3. PARAMETER CARDS

The program requires just the order card, with the following format:

Col. 1 to 4	Identification code: ●SPD
Col. 5	May contain one of the following values: - b̸ to duplicate the program file and print the program file index of the duplicated volume. - D if duplication only is desired. - P if print only is desired.

Col. 6 and 7	Logical number, punched in column 7, of the disc unit on which the program file indicated is being recorded; these columns remain blank if only the labels are to be printed (P punched in column 5).
Col. 8 to 72	Not used
Col. 73 to 80	Serial code (if desired).

4. DESCRIPTION AND EXECUTION

The program is made up of two segments.

It first checks the validity of the order card and, if program file duplication is desired, it checks that the volume involved in the duplication contains a file label in which the first two characters of the name are the same as on the file to be duplicated.

This label first has to be recorded by an earlier •FLW order.

The program then duplicates the file and prints the file index at the same time or either prints the index or duplicates the index only.

The following data is, however, always printed:

- volume label, free area table and original volume file label.
- if duplication is desired, the volume label, free area table and program file table of the disc on which the program file is being recorded. In such case, before printing the new file labels, the following message is printed:

  NEW DISC LABEL AND FILE LABEL

5. CHECKS AND ERROR MESSAGES

The program checks the operation of disc units and the validity of parameters punched in the order card.

When an error is found, it is indicated by one of the following messages:

0901 x D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or verifying logical disc unit N. The program terminates; feasibility condition: NO.

0902 x D. UNIT N OUT OF ORDER

Logical disc unit N signalled out of order. The program causes the paper on the printer to move to a new page and halts at location 01DA; pressing RUN retries the operation that caused the signal. If the signal repeats, the program terminates. Feasibility condition: NO.

0903 x D. UNIT N PERSISTENT ERROR

Positioning error on logical disc N. Program terminates; feasibility condition: NO.

0910 x ORDER CARD ERROR

Unrecognizable codes punched in column 5 of the order card. Program terminates; feasibility condition: NO.

0911 x DISC UNIT N CODE ERROR

Wrong data in column 6 & 7. This check is only carried out if there is a P punched in column 5.
Program terminates; feasibility condition: NO.

0912 x OVERFLOW OF PROGRAM FILE TN ... N LAST PROGRAM

The program file to be recorded on the volume indicated is greater than the area reserved for it.
Transcription of the file terminates; the last program to be recorded correctly was a type T named N ... N.
For an overflow during transcription of the subroutine library or the first program the following message is printed as the second line:

NEW FILE ZONE TOO SMALL

In the former case, the new program file can be used to the program indicated in the message; in the latter case, it cannot be used at all. The program terminates; feasibility condition: NO.

0913 x NEW PROGRAM FILE NOT AVAILABLE

When a new program file is being recorded, an error occurs which will prevent its future use. (e.g. error in recording the program file index). The program terminates; feasibility condition: NO.

0914 x UNIT N SD (LD) NOT FOUND ON PROGRAM FILE LABEL

The identification name of the file label of the volume on which recording is being carried out is not the same as the one on the volume to be duplicated. The program terminates; feasibility condition: NO.

0916 x UNIT N PROGRAM FILE LABEL NOT FOUND

The program file label is not found on the volume being recorded. The program stops; feasibility condition: NO.

6. EXAMPLE

To duplicate the program file of the library disc into the volume mounted on logical disc unit 3, and to print the program index and the subroutine library index at the same time, the order card should be punched as follows:

Col. 1 to 4	•SPD
Col. 5	ƀ
Col. 6 and 7	ƀ3
Col. 8 to 72	ƀ . . . ƀ
Col. 73 to 80	ƀ . . . ƀ

●SDC SYSTEM DISC CLOSE

1. FUNCTION

To delete from the system disc program file all programs following the program specified on the order card.

2. SYSTEM UNITS REQUIRED

The System Disc Close uses the following peripherals:

- a disc unit on which the System Disc is mounted
- a printer.

3. PARAMETER CARDS

System Disc Close requires only the order card with the following format:

Col. 1 to 4	Order identification code : ●SDC
Col. 5	Code indicating the type (1 or 3) of the last program to keep in the program file.
Col. 6 to 13	Name of the last program to keep.
Col. 14 to 72	Not used.
Col. 73 to 80	Serial code (if desired).

4. DESCRIPTION AND EXECUTION

The program is made up of a single segment located at address 2048 (decimal). It scans the program file directory for the record relative to the program specified in the order card and writes the character E (end of index) after it.

It then updates positions 60 to 64 of the program file label, which contain the address of the first free sector.

5. CHECKS AND ERROR MESSAGES

The program checks the operation of the disc units and the validity of parameters punched in the order card. When an error is detected, it is indicated by one of the following messages:

0601 * D. UNIT N PERSISTENT ERROR

Persistent error in recording, reading or verifying logical disc unit N.
Program terminates; feasibility condition NO.

0602 * D. UNIT N OUT OF ORDER

Logical disc unit N signalled out of order. The program causes the printer to move the paper to the next page and halts at location 01DA.
Pressing RUN retries the operation that caused the signal. If the signal repeats, the program terminates and the feasibility condition is NO.

0603 x D. UNIT N PERSISTENT ERROR

Positioning error on logical disc unit N.
Program terminates; feasibility condition: NO.

0610 x PROGRAM NAME NOT FOUND

The program specified in the order card is not found in the program file. The program terminates; feasibility condition: NO.

0611 x PROGRAM CODE ERROR

Type code punched in column 5 of the card is not 1 or 3.
Program terminates; feasibility condition : NO.

0698 x PROGRAM FILE NOT AVAILABLE

This message indicates that, as a consequence of the last message printed, the program file is not usable.

6. EXAMPLE

To close the SD after type 3 program whose name is AGGIOR, the order card should be punched as follows:

Col. 1 to 4	•SDC
Col. 5	3
Col. 6 to 13	AGGIOR.
Col. 14 to 80	b̸ . . . b̸

•CZD CYLINDER ZERO DUPLICATE

1. FUNCTION

This order copies Cylinder Zero of a System Disc onto another disc.

2. SYSTEM UNITS REQUIRED

- a printer
- two disc units, one for the SD, the other for the disc being updated.

3. PARAMETER CARDS

This program requires only the order card with the following format:

Col. 1 to 4	•CZD
Col. 5 to 6	logical unit number, of the disc being updated, punched in col. 6
Col. 7 to 72	not used
Col. 73 to 76	optional ID code
Col. 77 to 80	not used

4. DESCRIPTION AND EXECUTION

The program is madeup of a single segment.
It checks that the disc being updated was not declared on the same unit of the SD, then copies Cylinder Zero.

5. CHECKS AND ERROR MESSAGES

3601 * D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or verifiying the logical number of unit N.
Program execution stops at location 02A6.
Feasibility condition: NO.

3602 * D. UNIT N OUT OF ORDER

Disc unit N out of order. The program causes the printer to move the paper to a new page and halts at location 01DA. Pressing START retries the operation that caused the signal. If the signal repeats the program halts at location 02A6; feasibility condition:NO.

3603 * D. UNIT N PERSISTENT ERROR

Persistent positioning error on disc unit N.
Program execution stops at location 02A6; feasibility condition:NO.

3605* D. UNIT N NOT CONNECTED

The disc unit, whose number appears in col. 6, is not connected to the system.

3620 * DISC NUMBER ERROR

Wrong number punched in col. 6
Program execution stops at location 02A6.
Feasibility condition: NO.

3621 * OUTPUT DISC IS SYSTEM DISC

Logical unit number wrong: the number specified in col. 6 is the same assigned to the System Disc. Program execution stops at location 02A6; feasibility condition: NO.

3699 * END OF CYLINDER ZERO DUPLICATE

End of program.

6. EXAMPLE

For duplication of Cylinder Zero of an SD, mounted on unit 0 (zero), onto an output disc mounted on unit 2, punch the following card:

Col. 1 to 4	•CZD
Col. 5 to 6	ƀ2
Col. 7 to 80	Blank

•SDU SYSTEM DISC UPDATE

1. FUNCTION

To update and modify the SD in the following ways:

a) insert a new program

b) modify the label of a program

c) replace one or more program segment (s)

d) insert one or more segment (s) in a program

Programs may be of type 1 or type 3.

2. SYSTEM UNITS REQUIRED

The program uses the following peripherals:

- a disc unit on which the System Disc is mounted
- a reader to enter detail cards
- a printer.

3. PARAMETER CARDS

The program requires an order card and several detail cards.

3.1. ORDER CARD

The format of the order card is:

Col. 1 to 4	●SDU
Col. 5 to 72	Not used
Col. 73 to 80	Serial code.

3.2. DETAIL CARD PLC (Program Label Create)

This card generates a new program label. It is used whenever a new program is inserted in the SD or when an existing program is replaced.

The PLC card must, therefore, be followed by one or more SIN (Segment Insert) card(s) used to insert each segment of the program.

The PLC card also contains the name of the program to be inserted. This name will be added to the file index.

The format of the PLC card is as follows:

Col. 1 to 4	Identification code: ɓPLC
Col. 5	Program type code (1 or 3)
Col. 6 to 13	Program name
Col. 14	Code indicating the input/output routine used by the program. ɓ = program uses a tape management routine 1 = program uses a disc management routine 2 = program uses a tape and a disc management routine 3 = program uses neither tape nor disc management routine.

Col. 15 to 23	Program type code (1 or 3) and program name. These columns are used when the program name is to be changed.
Col. 24 to 27	Address of the file table, in coded hexadecimal. Punch 0000 if the file table is missing.
Col. 28 to 31	Starting address of the program, in coded hexadecimal.
Col. 32 to 35	Final address of the program plus one, (coded hexadecimal).
Col. 36	I - if the program is not included in the SD and must therefore be inserted. R - if a program with the same name, but which is to be replaced, is included in the SD. In this case, the existing program is deleted.
Col. 37 to 42	Date, expressed as ƀyyddd.
Col. 43 to 80	Not used.

3. 3. SIN (Segment Insert) DETAIL CARD

This card is used to insert a segment.

It is placed before the program cards for the segment to be inserted. If the next segment is part of a whole program which is newly created, a PLC card must be placed before the first SIN card. This is unnecessary, however, in the case of a segment which is already part of an existing program.

The format of the SIN card is as follows:

Col. 1 to 4	ƀSIN
Col. 5	Program type code (1 or 3)
Col. 6 to 13	Name of the program to which the segment belongs.

Col. 14 and 15	Number (decimal) of the program segment on the disc after which the segment made up of object program cards following this card is to be inserted. This number is not used for the SIN cards following this first deck, as any other segments will be inserted after the first one. This number is not used for the SIN cards following this first deck, as any other segments will be inserted after the first one. The first SIN card is numbered 00 when the first segment or a new program (PLC card with code I) is inserted.
Col. 16 to 19	Number (decimal) of object cards in the segment.
Col. 20 to 21	Number (decimal) assigned to the segment.
Col. 22 to 32	Not used.
Col. 33 to 36	Number (decimal) of cards making up the segment preceding the present card, if any
Col. 37 to 42	Date expressed as ƀyyddd
Col. 43 to 80	Not used.

3.3. SRP (Segment Replace) DETAIL CARD

The SRP card is used to replace an existing program segment. It is followed by program cards for the segment to be replaced.

The format of the SRP card is as follows:

Col. 1 to 4	ƀSRP
Col. 5	Code (1 or 3) of the program to be modified
Col. 6 to 13	Name of the program to be modified.
Col. 14 to 15	Number (decimal) of the segment to be replaced.

Col. 16 to 19	Number (decimal) of object program cards making up the segment.
Col. 20 to 32	Not used.
Col. 33 to 36	Number (decimal) of cards making up the segment preceding the present card, if any.
Col. 37 to 42	Date expressed as ƀyyddd
Col. 43 to 80	Not used.

3.4. END OF ORDER CARD

The end of the detail cards is indicated by the END card which is punched as follows:

Col. 1 to 4	ƀEND
Col. 5 to 32	Blank
Col. 33 to 36	Number (decimal) of cards making up the segment preceding the present card, if any.
Col. 37 to 42	Date expressed as ƀyyddd
Col. 43 to 80	Not used.

4. DESCRIPTION AND EXECUTION

This program is made up of two segments located at memory address 2048 (decimal).

The program may be used to insert a new program in the SD, to modify or replace part of an existing program in the SD.

In the former case, segment blocks have a maximum length of 1024 characters and are inserted at the end of the program file, which may mean deletion of type C programs.

In the latter case, the program carries out the following operations in sequence:

- insertion of a new program at the end of the program file,
- insertion of the program name, at the end of the file index, and deletion of type C programs from the index,
- deletion of the former program name from the index when an existing program is replaced,
- trascription of the entire file, at the same address, so that sectors taken up by the deleted program are again occupied, and then reorganization of the linking between pages and labels.

The date punched on the first detail card that has been read or, if no date is punched, the day's date, is always written in the program label.

5. CHECKS AND ERROR MESSAGES

1301 x D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or verifying logical disc unit N. The program stops; feasibility condition: NO.

1302 x D. UNIT N OUT OF ORDER

Logical disc unit N reported as out of order. The program causes the printer to slew the paper to a new page and the program halts at location 01DA. Pressing RUN retries the operation that caused the message. If the message repeats, the program stops. Feasibility condition: NO.

1303 x D. UNIT N PERSISTENT ERROR

Persistent positioning error on logical disc unit N. Program terminates; feasibility condition: NO.

1310 x SUB ORDER CARD ERROR

Wrong detail card read. Program terminates; feasibility condition: NO.

1311 x SUB ORDER CARD COLUMN NN ERROR

The area starting at columns NN of the previously printed is wrong. Program terminates; feasibility condition: NO.

1312 x MULTIPLE PROGRAM NAME

Request to insert a program (PLC card with code I) which is already included in the program index.
Program continues; feasibility condition: NO.

1313 x PROGRAM NOT FOUND

Program to be updated is not included in program file.
Program terminates; feasibility condition: NO.

1314 x OVERFLOW OF PROGRAM TABLE

The number of programs in the SD index has reached its maximum of 259. Program terminates; feasibility condition: NO.

1315 x SUB-ORDER CARD SEQUENCE ERROR

Detail cards not inserted in correct order. Program terminates; feasibility condition: NO.

1317 x CARD OF OBJECT PROGRAM ERROR

Incorrect program card read for insertion. The program stops; feasibility condition: NO.

1318 x SEGMENT NUMBER ERROR

Incorrect error segment number punched in SIN or SRP cards.
The program stops: feasibility condition: NO.

1319 x PROGRAM CODE ERROR

Incorrect program code is card to be inserted. The program stops; feasibility condition: NO.

1320 x CARD COUNTER ERROR

Number of cards making up a segment not the same as the number of cards specified in the SIN or SRP card.
If the number involved is the one punched in cols. 16-19, the program continues.
Feasibility condition: YES.
If the number involved is the one punched in cols. 33-36 the program stops.
Feasibility condition: NO.

1321 x MAX ADDRESS ERROR

The maximum memory address of the program does not agree with the address punched in the program label. The program continues.
Feasibility condition: YES.

1322 x SEGMENT NOT FOUND

The program has no segment with the number indicated in the SRP or SIN card. The program stops. Feasibility condition: NO.

1323 x OVERFLOW OF PROGRAM FILE TN . . . N LAST PROGRAM

At the end of recording program file size exceeds that of the area reserved for it.
The program stops and the last program that was inserted correctively on the SD is named N . . . N and is a type T program.
Feasibility condition: NO.

1398 x PROGRAM FILE NOT AVAILABLE

This message indicates that as a consequence of the last message printed, the program file can no longer be used.

6. EXAMPLE

To insert on the SD, a type 3 program with code 4576 and name FASE 1, which is made up of a single segment and requires only card read and write subroutines, start address 2048, end address of the program (containing 100 cards) 4095, the following three cards should be punched as indicated if the day's date is to be written in the program label:

Other Card:

Col. 1 to 4	•SDU
Col. 5 to 80	Blank

Detail Card PLC:

Col. 1 to 4	b̸PLC
Col. 5	3
Col. 6 to 13	FASE1
Col. 14	3
Col. 15 to 23	Blank
Col. 24 to 27	0000
Col. 28 to 31	0800
Col. 32 to 35	1000
Col. 36	I
Col. 37 to 80	Not used

Detail Card SIN

Col. 1 to 4	b̸SIN
Col. 5	3
Col. 6 to 13	FASE1
Col. 14 and 15	00
Col. 16 to 19	0100
Col. 20 and 21	01
Col. 22 to 80	Not used.

These three cards should be inserted in the order listed and should be followed by the program card for the program to be generated. The last card should be an END card. The format of this card should be as follows if the actual number of cards making up the segment is to be checked the one punched in the END card:

Col. 1 to 4	b̸END
Col. 5 to 32	Not used
Col. 33 to 36	0100
Col. 37 to 80	Not used.

• SDS SYSTEM DISC STRIP

1. FUNCTION

To delete type 1 or 3 program from the System Disc.

2. SYSTEM UNITS REQUIRED

To use this program, the following peripherals are needed:

- a disc unit on which the SD is installed
- a printer
- a card reader.

3. PARAMETER CARDS

The System Disc Strip requires an order card, one or more detail card(s) and an END card.

3.1. ORDER CARD

Col. 1 to 4	order identification code •SDS
Col. 5 to 72	not used
Col. 73 to 80	serial code.

3.2. DETAIL CARD SDS

Col. 1 to 4	identification code ƀSDS
Col. 5	code indicating the type of program to be deleted (1 or 3)
Col. 6 to 13	name of program to be deleted
Col. 14	code C if the program to be deleted is a type C program. Blank otherwise.
Col. 15 to 80	Not used.

3.3. END CARD

The end of detail cards is signalled by the END card.

4. DESCRIPTION AND EXECUTION

The program is made up of a single segment located at memory address 2048 (decimal).

The program deletes all programs indicated in the detail cards from the SD program index.

At the end, either the index or the entire program file is rewritten to fill in blank spaces, and the file label is updated in keeping with the new occupation limits.

5. CHECKS AND ERROR MESSAGES

The program checks the operation of peripheral unit used and the validity of parameters punched in the cards.

When an error is detected, it is indicated by one of the following messages:

1601 x D. UNIT N PERSISTENT ERROR

Persistent reading, writing or verifying error on logical disc unit N.
Program terminates; feasibility condition: NO.

1602 x D. UNIT N OUT OF ORDER

Logical disc unit N signalled out of order. The program causes the printer to move the paper to a new page and halts at location 01DA.
Pressing RUN retries the operation that caused the signal. If the signal repeats the program terminates.
Feasibility condition: NO.

1603 x D. UNIT N PERSISTENT ERROR

Persistent positioning error on logical disc unit N. Program terminates; feasibility condition: NO.

1610 x PROGRAM N . . . N NOT FOUND

The requested program is not found in the program type. Program continues normally; feasibility condition at the end of execution is: NO.

1611 x PROGRAM CODE ERROR

Wrong punch in column 5 or 14 of a detail card. Remove the card and processing continues normally: at the end of processing, feasibility condition is: NO.

1612 x SUB-ORDER CARD ERROR

An unrecognizable card is read or the END card comes immediately after the order card.
Processing continues normally: feasibility condition at the end of processing is: NO.

1698 x PROGRAM FILE NOT AVAILABLE

This message indicates that, as a consequence of the last message printed, the program file is not usable.

6. EXAMPLE

To delete all type 3 programs from the SD, the following cards should be punched:

Order Card

Col. 1 to 4	•SDS
Col. 5 to 80	Blank.

Detail Card

Col. 1 to 4	ƀSDS
Col. 5	3
Col. 6 to 13	Blank
Col. 14	ƀ
Col. 15 to 80	Blank

END Card:

Col. 1 to 4	ƀEND
Col. 5 to 80	Blank.

•SLS SUBROUTINE LIBRARY STRIP

1. FUNCTION

To delete one or more subroutines from the System Disc subroutine library.

2. SYSTEM UNITS REQUIRED

The following peripherals are used by this program:

- a disc unit on which the System Disc is installed,
- a card reader to enter detail cards,
- a printer.

3. PARAMETER CARDS

The Subroutine Library Strip program uses an order card and one or more detail cards.

3.1. ORDER CARD

The format of the order card is:

Col.	1 to 4	Order identification code: •SLS
Col.	5 to 72	Not used
Col.	73 to 80	Serial code

3.2. DETAIL CARD SLS

Col. 1 to 4	Identification code: ƀSLS
Col. 5 to 10	Name of the subroutine to be eliminated
Col. 11 to 80	Not used.

3.3. END OF ORDER CARD

Col. 1 to 4	ƀEND
Col. 5 to 80	Not used.

4. DESCRIPTION AND EXECUTION

The program is made up of a single segment located at memory address 2048 (decimal). First, all detail cards are read and printed and the new subroutine library directory is generated in the work area. Should an error occur during this phase of the operation, the program terminates and the program file returns to its initial state. The new subroutine library index is then inserted in place of the former one and, finally, the entire program file is again inserted and fills the area freed by the deleted subroutines.

An error during the latter phase may make the file unusable; at the end of the phase, the names of the last program and subroutine entered are printed and the file is closed in the normal manner. And, finally, the program prints the new subroutine library index after the message: NEW INDEX OF SUBROUTINE LIBRARY.

5. CHECKS AND ERROR MESSAGES

The program checks the validity of detail cards and the operation of peripherals used, in the event of an error, one of the following messages is printed:

3101 x D. UNIT N PERSISTENT ERROR

Persistent reading, writing or verifying error on logical disc unit N.
Program terminates, feasibility condition: NO.

3102 x D. UNIT N OUT OF ORDER

Logical disc unit N signalled out of order.
The program causes the printer to move the paper to a new page and halts at location 01DA. Pressing RUN retries the operation that caused the signal. If the signal repeats, the program terminates; feasibility condition: NO.

3103 x D. UNIT N PERSISTENT ERROR

Persistent positioning error on logical disc unit N. Program terminates; feasibility condition: NO.

3110 x SUB-ORDER CARD ERROR

Error in the format of the detail card: the first four columns do not contain ␢SLS
The program terminates; feasibility condition: NO.

3111 x SUBROUTINE XXXXXX NOT FOUND

Subroutine XXXXXX specified in the detail card cannot be found in the SD library.
Program continues normally; feasibility condition: NO.

3197 x NO WORK AREA ON SD

The SD has no work area, which is needed to generate the new subroutine table and to transcribe the file, or the work area on the SD is not extensive enough. Program terminates; feasibility condition: NO.

3198 x PROGRAM FILE NOT AVAILABLE

This message indicates that, as a consequence of the last message printed, the program file is no longer usable.

6. EXAMPLE

To delete the two subroutines YDIVS and YRD40 from the SD library, one order card and two detail cards must be punched as follows:

Order Card:

Col. 1 to 4	•SLS
Col. 5 to 80	Blank.

Detail Card:

1)	Col. 1 to 4	␢SLS
	Col. 5 to 10	YDIVS
	Col. 11 to 80	Blank
2)	Col. 1 to 4	␢SLS
	Col. 5 to 10	YRD40
	Col. 11 to 80	Blank.

END Card:

Col. 1 to 4	␢END
Col. 5 to 80	Blank.

•SLI SUBROUTINE LIBRARY INSERT

1. FUNCTION

To insert a new subroutine at the end of the System Disc subroutine library. This operation is carried out in three distinct phases:

a) Standardization of the subroutine to conform with the specific format of DOS subroutines;

b) Insertion of the subroutine in the System Disc library

c) Updating the subroutine library index.

2. SYSTEM UNITS REQUIRED

The following peripherals are needed to operate this program:

- a logical disc unit on which the System Disc is installed;
- a printer.

3. PARAMETER CARDS

Subroutine Library Insert uses only the order card with the following format:

Col. 1 to 4	Order identification code: •SLI
Col. 5 to 10	Name of the subroutine to be inserted
Col. 11	- 1, if the subroutine forms part of EIOS, excluding YOPEND, YCLOSD, YNEXTD, YDCKP. - blank, if not.
Col. 12 to 72	Not used
Col. 73 to 80	Serial code (if desired).

4. DESCRIPTION AND EXECUTION

The APS II compiler supplies the subroutine with three-sector fixed length pages written in the 6-bit mode. Only 276-character pages are used.

Each page contains two logical records with a fixed length of 138 characters each.

The first 80 positions of each record contain the original statement is source language or statements generated by a macro-instruction.

Positions 81 to 110 contain blanks or data which do not concerne the object program.

Positions 111 to 119 may contain error messages; positions 120 to 138 contain the following data, in pure unpacked binary:

120 & 121	length of the statement (nn-1 for constants and areas)
122 & 125	statement address
126 to 127	operation code
128 & 129	complement of operation (duplication factor for constants and areas)
130 to 133	address of the first operand or zeros (decimal)
134 to 137	address of the second operand or zeros (decimal)
138	0(decimal)

During the first phase, basic language statements are organized into pages having a maximum length of 1200 characters and are set as described above.

Constants of the DC'C' type are handled as follows:

- they are standardized with spaces to the right when the effective length is shorter than specified;

- they are unpacked with a quartet;
- they are duplicated using a duplication factor.

Constants of the DC'X' type are handled as follows:

- they are standardized with spaces to the right when the effective lenght is shorter than indicated (nn)
- they are unpacked with a zone quartet
- when the duplication factor is greater than 1, the standardized and unpacked constant is repeated the number of times indicated by the duplication factor.

Constants of the DC"A" type, however, are considered as statements. DS and ORG statements close pages being formulated; data for identifying the subroutine are written at the beginning of each page.

During the second phase, the subroutine is inserted in the library, after the end of the existing subroutines, and the library index is updated.

Type 3 programs generated by the order •DPG are deleted.

5. CHECKS AND ERROR MESSAGES

The program checks the operation of peripherals and the validity of the order card; it also checks the organization of the subroutine and indicates errors by means of one of the following messages:

3001 x D. UNIT N PERSISTENT ERROR

Persistent reading, writing or verifying error on logical disc unit N.
Program terminates; feasibility condition: NO.

Statements *, LF, EJEC are ignored with the exception of statement *, read immediately before the END statement.
In the card relating to the * statement, the user will have punched the subroutine's generation date (in the form ȸYYDDD) in columns 31 - 36. This date will be recorded in positions 8 - 13 of each subroutine block in its final form.

6. RESTRICTIONS

The subroutines processed by the order oSLI have the following restrictions:

- out-of-sequence statements are not admitted
- in AL1 type DC statements a symbolic address is not admitted
- DC statements with continuation are not admitted
- in ORG statements the only operand admitted is R. In this case, the subroutine's name should begin and end with the letter Y
- the statement RORG, ESCT, ESGM are not admitted
- the name of another subroutine is not admitted as an instruction operand. The only exception is when another subroutine is called by a JRT or JU instruction. In this case not that the subroutine called must be present in the library; also, the •APS program prints a U signal exclusively to show that a jump has been made to another subroutine
- adjustment is not admitted if the operand of a JU or JRT statement is the name of a called subroutine.

3015 * INVALID SUBROUTINE: INSERTION IMPOSSIBLE

A subroutine is about to be inserted: but a subroutine bearing the same name is already present in the library.
Program execution is interrupted.
Feasibility condition: NO.

3098 * PROGRAM FILE NOT AVAILABLE

This message indicates that as a consequence of the last message printed, the program file is not usable.

3099 * END OF SUBROUTINE LIBRARY INSERT

End of program. The order has been correctly executed.

6. EXAMPLE

To insert the subroutine YMULN, which has just been compiled, the order card should be punched as follows:

Order Card SLI

Col. 1 to 4	●SLI
Col. 5 to 10	YMULNb̸
Col. 11 to 80	Blank.

3002 x D. UNIT N OUT OF ORDER

Logical disc unit N signalled out of order. The program causes the printer to move the paper to a new page and halts at location 01DA. Pressing RUN retries the operation that caused the signal. If the signal repeats, the program terminates. Feasibility condition: NO.

3003 x D. UNIT N PERSISTENT ERROR

Positioning error on logical disc unit N.
Program terminates; feasibility condition: NO.

3010 x INVALID SUBROUTINE: INSERTION IMPOSSIBLE

Name of the subroutine as read on the order card not the same as the one in the work area.
Program terminates; feasibility condition: NO.

3011 x INVALID SUBROUTINE: INSERTION IMPOSSIBLE

Storage address of the subroutine other than 0. Program terminates; feasibility condition: NO.

3012 *INVALID SUBROUTINE: INSERTION IMPOSSIBLE

Error in the subroutine statements; program terminates; feasibility condition: NO.

3013 x INVALID SUBROUTINE: INSERTION IMPOSSIBLE

Subroutines requested cannot be found in the library. Program terminates; feasibility condition: NO.

3014 x OVERFLOW OF PROGRAM FILE INSERTION IMPOSSIBLE

Program file cannot hold the subroutine. Program terminates; feasibility condition: NO.

•SPC SUBROUTINE PERIPHERAL COMPILE

1. FUNCTION

To change the names of peripheral units contained in certain specific subroutines within the System Disc Library.

2. SYSTEM UNITS REQUIRED

This program uses the following peripherals:

- a disc unit on which the System Disc is installed;
- a card reader for entering detail cards;
- a printer.

3. PARAMETER CARDS

The Subroutine Peripheral Compile program uses an order card and one or more detail card (s).

3.1. ORDER CARD

Col. 1 to 4	Order identification code: •SPC
Col. 5 to 72	Not used
Col. 73 to 80	Serial code (if desired)

3.2. DETAIL CARD SPC

Col. 1 to 4	Identification code: ɃSPC

Col. 5 to 10	Name of the subroutine to be modified
Col. 11 to 13	Name of the peripheral unit being changed.
Col. 14 to 16	New peripheral unit name
Col. 17 to 19	Second peripheral unit name for unit being changed (if desired)
Col. 20 to 22	New second peripheral unit name
Col. 23 to 80	Not used.

If there is no second peripheral unit to change, columns 17 and 22 are not used.

3.3. END OF ORDER CARD

Col. 1 to 4	ƀEND
Col. 5 to 80	Not used

4. DESCRIPTION AND EXECUTION

The program is made up of a single segment located at memory address 2048 (decimal).

The program reads, checks and prints a single detail card at a time; for each detail card, the program looks for the specified subroutine and changes the name of the peripheral unit with a new name indicated on the cards.

Characters within the subroutine to be changed are located in area "B", which is reserved for peripheral unit names.

5. CHECKS AND ERROR MESSAGES

The program checks the peripheral units used as well as the validity of punches in the parameter cards; in the event of error, one of the following messages is printed:

3201 * D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or verifying logical disc unit N.
Program terminates; feasibility condition: NO.

3202 * D. UNIT N OUT OF ORDER

Logical disc unit N signalled out of order. The program causes the printer to move the paper to a new page and halts at location 01DA. Pressing RUN retries the operation that caused the signal. If the signal repeats, the program terminates; feasibility condition: NO.

3203 * D. UNIT N PERSISTENT ERROR

Persistent positioning error on logical disc unit N. Program terminates; feasibility condition: NO.

3210 * SUB-ORDER CARD ERROR

b̸SPC not punched in the first four columns of the detail card.
Program terminates; feasibility condition: NO.

3211 * SUBROUTINE XXXXXX NOT FOUND

Subroutines specified on the detail card not found on the SD library. Program continues normally; at the end of execution, feasibility condition: NO.

3212 * PERIPHERALS XXX NOT FOUND

Peripheral unit XXX not found in the subroutine specified on the last detail card. Program continues normally; at the end of execution; feasibility condition: NO.

3213 x SUB ORDER CARD COLUMN NN ERROR

Indicates that the area beginning with column nn on the detail card is wrong. Program continues normally; at the end of execution, feasibility condition is: NO.

3298 x PROGRAM FILE NOT AVAILABLE

This message indicates that, as a consequence of the last message printed, the program file is no longer usable.

6. EXAMPLE

To change keypunch 048 in subroutine YP85 so that it is replaced by the number 064, the following parameter cards should be punched:

Order card:

Col. 1 to 4	•SPC
Col. 5 to 80	Blank.

Detail card:

Col. 1 to 4	ƀSPC
Col. 5 to 10	YP85
Col. 11 to 13	048
Col. 14 to 16	064
Col. 17 to 80	Blank.

These are followed by the END card.

●LDU LIBRARY DISC UPDATE

1. FUNCTION

This program allows:

- Generation of a new Library Disc using programs taken from a System Disc or Library Disc.
- Updating of a Library Disc using a System Disc or another Library Disc as the basic disc.
- In this case, new programs may be inserted or existing programs may be deleted in the Library Disc, while the file tables of the new and existing programs are updated.
- Updating of a Library Disc using the same Library Disc as the basic disc. In this case only obsolete programs may be eliminated or the file tables may be altered.

2. SYSTEM UNITS REQUIRED

The following peripherals are needed to use this program:

- two disc units, one for the basic system disc and the other for the new Library Disc (a single unit may be used if the basic disc is the Library Disc to be updated).
- a card reader for detail card input.
- a printer.

3. PARAMETER CARDS

This program requires an order card and a number of detail cards followed by an END card.

3.1. THE ORDER CARD

Col. 1 to 4	Identification code:•LDU
Col. 5 and 6	Logic device number (b̸0 - b̸9, or WD) on the unit on which the volume that is to contain the new Library Disc is to be mounted if two disc units are used. - LDcode, if only one disc unit is used on which the Library Disc is mounted (as in the case of updating the Library Disc used also as basic disc)
Col. 7 to 23	Name of the program file processed, in the case of updating, or name of the new file in the case of generation.
Col. 24	Code specifying the type of work to be done. It may assume either of two values: * (asterisk), if a new Library Disc is being generated, b̸ (blank) if updating.
Col. 25 to 72	Not used.
Col. 73 to 80	Serial code.

3.2. THE PRM DETAIL CARD (Program Remove)

This card is used to delete a program from the Library Disc.

Col. 1 to 4	Identification code: b̸PRM
Col. 5	Code of the program to be eliminated (1 or 3)

Col. 6 to 13	Name of the program to be eliminated
Col. 14 to 80	Not used.

The PRM cards must immediately follow the order card.

3.3. THE PRI (Program Insert) DETAIL CARD

This card is used to insert a program.

Col. 1 to 4	Identification code: ƀPRI
Col. 5	Code of the program to be inserted (1 or 3)
Col. 6 to 13	Name of program forming part of basic disc to be inserted in Library Disc.
Col. 14 to 80	Not used.

3.4. THE PRT (Program Transfer) DETAIL CARD

This card is used to update the program file table.

Col. 1 to 4	Identification code: ƀPRT
Col. 5	Code of the program to be modified (always 3)
Col. 6 to 13	Number of the program forming part of the Library Disc whose file table is to be updated using the LBC card · following the PRT card
Col. 14 to 80	Not used.

Each PRT card is followed by as many LBC cards (see Appendix, page 310) as there are files to be processed.

3.5. THE END-OF-ORDER CARD

The end of the detail cards is indicated by an END card.

4. DESCRIPTION AND EXECUTION

The program consists of four segments with the same memory address of 1800 (decimal).

It can perform one of the following functions :

a) Delete programs from the Library Disc.

b) Update the file tables of the Library Disc program.

c) Insert new programs taken from the System Disc or the Library Disc with or without modifying their file tables.

d) Generate a new Library Disc from the System Disc or from another Library Disc.

If the order •LDU requires a single disc unit, only the LD code is present in cols. 5 - 6 of the order card and a check is carried out to make sure that the volume is a Library Disc. In this case the Library Disc can be updated only by means of the PRM and PRT cards.

If there is a code other than LD in cols. 5 - 6 of the order card (i. e. ⱺ0 - ⱺ9 or WD), the basic disc may be either a System Disc or a Library Disc, while the second volume must be a Library Disc.

If a new Library Disc is being generated (x in col. 24 of the order card), the program accepts only PRI cards; for updating (ⱺ in col. 24 of the order card) all types of detail cards are accepted except when a single disc unit is used (LD code in cols. 5 - 6 of the order card); in this case, as has already been said, only PRM and PRT cards may be used.

5. CHECKS AND ERROR MESSAGES

The program checks the parameter cards and peripheral units involved. Any error will result in one of the following messages:

1501 x D. UNIT N PERSISTENT ERROR

A persistent error in writing, reading or verifying has been encountered on the disc unit with logical number N. The program stops. Feasibility condition: NO.

1502 x D. UNIT N OUT OF ORDER

The unit with logical number N is out of order. The program slews the paper to a new page and comes to a HLT at address 01DA.
The operation causing the message is retried by pressing RUN.
If the message is repeated the program stops.
Feasibility condition: NO.

1503 x D. UNIT N PERSISTENT ERROR

A persistent positioning error has been encountered on the disc unit with logical number N. The program stops. Feasibility condition: NO.

1510 x SUB-ORDER CARD ERROR

An undentifiable card has been read. The program stops.
Feasibility condition: NO.

1511 x SEQUENCE ERROR SUB-ORDER CARDS

A detail card has been found out of sequence. The program stops.
Feasibility condition: NO.

1512 x LBC-CARD NOT FOUND

A PRT card has been read which was not followed by an LBC card.
The program stops. Feasibility condition: NO.

1513 x D. UNIT N : PROGRAM FILE LABEL NOT EXACT

The program file label is not the one required. It is printed after reading, then the program stops.
Feasibility condition: NO.

1514 x ORDER CARD COLUMN NN ERROR

Illegal coding in cols. NN of the order card. The program stops. Feasibility condition: NO.

1515 x D. UNIT N : PROGRAM FILE LABEL NOT FOUND

The volume mounted on the disc unit with logical number N has no program file label. The program stops.
Feasibility condition: NO.

1516 x PROGRAM T N . . . N NOT FOUND

Program T, name N . . . N is missing in program file list. The program stops. Feasibility condition: NO.

1517 x MULTIPLE PROGRAM NAME

The program whose insertion has been requested is already present on the Library Disc. The new program is inserted the same.
Program execution continues.
Feasibility condition: YES.

1518 x SUB-ORDER COLUMN NN ERROR

The filed starting from col. NN of the card is incorrect. Once the card has been read the program stops.
Feasibility condition: NO.

1519 x D. UNIT N : END OF PROGRAMS TABLE NOT FOUND

The closing character has not been found after reading a page of the program table.
The program wanted has not been found.
Program execution stops.
Feasibility condition: NO.

1520 x OVERFLOW OF PROGRAM FILE N . . . N LAST PROGRAM

The zone reserved for the Library Disc is not large enough to contain all the programs to be inserted. The last regularly inserted program has the name N . . . N. The program stops.
Feasibility condition: NO.

1521 x NO FILE TABLE

The program present on the Library Disc to be updated and whose file table is to be altered using the PRT card has no file table. The program stops. Feasibility condition: NO.

1598 x PROGRAM FILE NOT AVAILABLE

Indicates that as a result of the last error encountered, the program file can no longer be used. The program stops. Feasibility condition: NO.

6. EXAMPLE

The program with code 4002 is to be deleted from the Library Disc, the program with code 7004 contained in the SD is to be added and the file table of the program with code 8521 contained in the LD is to be updated. The disc units involved have the following logical number: LD, 0; SD, 1. The following cards must be punched:

1. ORDER CARD

Col. 1 to 4	•LDU
Col. 5 and 6	b̸0
Col. 7 to 23	LDK 115
Col. 24	b̸
Col. 25 to 80	b̸ . . . b̸

2. PRM DETAIL CARD

Col. 1 to 4	b̸PRM
Col. 5 to 12	4002
Col. 13 to 80	b̸ . . . b̸

3. PRT DETAIL CARD

Col. 1 to 4	b̸PRI
Col. 5	3
Col. 6 to 13	7004
Col. 14 to 80	b̸ . . . b̸

4. PRT DETAIL CARD

Col. 1 to 4	b̸PRT
Col. 5	3
Col. 6 to 13	8521
Col. 14 to 80	b̸ . . . b̸

The latter card must be followed by the corresponding LBC cards.

5. END card

Col. 1 to 4	b̸END
Col. 5 to 80	b̸ . . . b̸

•MDG

1. FUNCTION

To generate a Master Disc.

Program making up the Master Disc are taken from the Library Disc or the System Disc.

The file on the Master Disc has the following characteristics:

- it contains the Monitor program, which automatically links the programs;
- at the end of each program, a segment 99 is inserted; this segment contains all the indications needed by the Monitor to select the next program;
- a segment 0 is inserted at the beginning of each program; this segment contains indications for starting up the program referred to. Segment 0 functions are in accordance with the options specified in UPC detail cards.
- it always contains program •INT and •PRL.

The function of segment 99 is to load the Monitor program and to provide indications for selecting the new program to be started. The Monitor functions in accordance with the options specified in detail card PRE.

2. SYSTEM UNITS REQUIRED

The following peripherals are needed to use this program:

- a disc unit for the basic disc
- a disc unit for the volume to be inserted

- a disc unit for the System Disc or the Library Disc, if the program file to be written on the Master Disc is on a volume other than the basic disc
- a card reader for detail card input
- a printer.

3. PARAMETER CARDS

The program requires the use of an order card and several detail cards, followed by an END card.

3.1. THE OTHER CARD

Col. 1 to 4	Identification code : ●MDG
Col. 5 to 6	logical number (b̸0 - b̸9) of the unit on which the volume containing the program file to be written on the Master Disc is mounted. SD if the file is the System Disc, LD if it is the Library Disc
Col. 7 to 8	logical number (b̸0 - b̸9, or WD) of the unit that is to contain the Master Disc
Col. 9 to 25	name of the file to be assigned to the MD; the first two characters must be MD
Col. 26 to 72	not used
Col. 73 to 80	serial code.

3.2. OPC (Operating System Program Call) DETAIL CARD

This card is used to insert a type-1 program on the MD.

Col. 1 to 4	Identification code: ƀOPC
Col. 5 to 12	name of program to be inserted
Col. 13 to 80	not used

3.3. UPC (User Program Call) DETAIL CARD

This card is used to insert a type-3 program on the MD and to specify the corresponding start conditions.

Col. 1 to 4	Identification code: ƀUPC
Col. 5 to 12	program name
Col. 13 and 14	not used
Col. 15	1st preparation code for segment 0: - H if there is a halt before start up of the requested program - ƀ otherwise
Col. 16	2nd preparation code for segment 0: - * if 0A is to be stored in the memory before starting the program - ƀ otherwise
Col. 17	3rd preparation code for segment 0: - * if the characters specified in cols. 18 - 49 are to be inserted in the user data area - ƀ otherwise

Col. 18 to 49	user data (if desired)
Col. 50 to 80	not used.

3.4. PRE (Program End) DETAIL CARD

This card is used to put on the MD the linkage conditions for the next program.

Between the UPC and PRE cards the user can insert the corresponding LBC cards if the program has a file table.

Col. 1 to 4	Identification code: ƀPRE
Col. 5 to 12	name of program involved
Col. 13	Preparation code for the Monitor. It may assume one of the following values: 1 = start up of a specific type-3 program, selected during MD generation 2 = start up of a type-3 program, selected from among two programs specified during MD generation, depending on how SWITCH 2 is positioned 3 = read an order card and start Pilot 0 to execute the order. 4 = stop by •MTS 5 = start a program selected by the preceding program. The number of this program is located in card read area.
Col. 14 to 21	Name of type-3 program to be started, if code 1 is punched in column 13. Name of type-3 program to be started if SWITCH 2 is not set. Only applicable if code 2 is punched in column 13.

Col. 22 to 29	Name of program to be started if SWITCH 2 is set. Only applicable if code 2 is punched in col. 13.
Col. 30 to 80	Not used.

4. DESCRIPTION AND EXECUTION

The program consists of three segments: the second segment constitutes the order •PRL (which also acts as Monitor). This order is inserted in the Master Disc program file.

The first segment checks the order card and puts the orders •PRL, •INT and •MDS on the Master Disc to be generate.

The third segment reads the detail cards and inserts the user orders or programs specified on these cards.

5. CHECKS AND ERROR MESSAGES

The program checks the operation of peripheral units involved and the validity of the parameters punched on cards.

When an error is encountered, one of the following messages is printed:

1701 x D. UNIT N PERSISTENT ERROR

A persistent read, write or check error has been encountered on disc unit with logical number N.
The program stops. Feasibility condition: NO.

1702 x D. UNIT N OUT OF ORDER

Unit with logical number N is out of order.
The program slews paper to the next page and halts. Pressing RUN retries the operation that caused the message. If the message is repeated the program stops. Feasibility condition: NO.

1709 x D. UNIT N PERSISTENT ERROR

A persistent positioning error has been found on disc unit with logical number N.
The program stops.
Feasibility condition: NO.

1710 x SUB-ORDER CARD ERROR

An unidentifiable card has been read. The program stops.
Feasibility condition: NO.

1711 x SUB-ORDER CARD COLUMN NN ERROR

The field starting from col. NN of previously read card, is incorrect. The program stops.
Feasibility condition: NO.

1712 x D. UNIT N : PROGRAM FILE LABEL NOT FOUND

The program file label has not been found on disc N. The program stops. Feasibility condition: NO.

1713 x SEQUENCE ERROR SUB-ORDER CARDS

The detail cards area read in incorrect sequence. The program stops. Feasibility condition: NO.

1714 x PROGRAM TN ... N NOT FOUND

Requested type T program named N not included in program file. The program ignores the card and continues normally.

1715 x OVERFLOW OF PROGRAM FILE TN ... N LAST PROGRAM

The program file is larger than the size reserved for it. The last regularly written program is a type T program with name N ... N. The program stops. Feasibility condition: NO.

1716 x ORDER CARD COLUMN NN ERROR

The field starting from column NN is incorrect. The program stops. Feasibility condition: NO.

1717 x D. UNIT N : PROGRAM FILE LABLE NOT EXACT

The program file label is not the one expected. The label that has been read is printed.
Feasibility condition: NO.

6. EXAMPLE

A Master Disc is to be generated on disc unit 1. The following programs are taken from the Library Disc:

- FLW (type 1) and 4250 4270 4271 (type 3). The programs are to be started consecutively, with no interruptions in between; when the operation is completed the "end-of-job" stop will be executed. The following parameter cards must be punched:

1. Order Card

Col. 1 to 4	•MDG
Col. 5 to 6	LD
Col. 7 and 8	ƀ1
Col. 9 to 25	MDESEMPIO
Col. 26 to 80	Not used.

2. Detail card OPC

Col. 1 to 4	ƀOPC
Col. 5 to 8	•FLW
Col. 9 to 80	Blank

3. First Detail card UPC

Col. 1 to 4	b̸UPC
Col. 5 to 12	4250
Col. 13 & 14	Blank
Col. 15	b̸
Col. 16	b̸
Col. 17	b̸
Col. 18 to 80	Blank

4. Corresponding PRE card

Col. 1 to 4	b̸PRE
Col. 5 to 12	4250
Col. 13	1
Col. 14 to 21	4270
Col. 22 to 80	Blank

5. Second detail card UPC

Col. 1 to 4	b̸UPC
Col. 5 to 12	4270
Col. 13 & 14	Blank
Col. 15	b̸
Col. 16	b̸
Col. 17	b̸
Col. 18 to 80	Blank

6. Corresponding PRE card

Col. 1 to 4	ƀPRE
Col. 5 to 12	4270
Col. 13	1
Col. 14 to 21	4271
Col. 22 to 80	Not used.

7. Third detail card UPC

Col. 1 to 4	ƀUPC
Col. 5 to 12	4271
Col. 13 to 14	Blank
Col. 15	ƀ
Col. 16	ƀ
Col. 17	ƀ
Col. 18 to 80	Blank

8. Corresponding PRE card

Col. 1 to 4	ƀPRE
Col. 5 to 12	4271
Col. 13	4 (end of job)
Col. 14 to 80	Blank

9. END card

Col. 1 to 4	ƀEND
Col. 5 to 80	Not used

When loading the card reader, all cards must be in strict sequence, as described above.

•MDS MASTER DISC STOP

1. FUNCTION

To stop Master Disc processing.

2. PARAMETER CARD

The parameter card format is as follows:

Col. 1 to 4	Identification code: •MDS
Col. 5 to 72	Not used
Col. 73 to 80	Serial code if required.

3. EXECUTION

The printer makes a slew to the next page, then the program stops at address 01D6. Pilot 0 is restarted by pressing RUN.

4. MESSAGES

Processing of the order causes, the following message to be printed:

7601 x MASTER DISC STOP

Processing halts at address 01D6.
Pilot 0 is restarted by pressing RUN.

•DDP DISC DUMP

1. FUNCTION

To print the part of the file found on the specified volume, or to print the entire file.
First the volume label is printed then the program file label is printed, then the program file is printed.

2. SYSTEM UNITS REQUIRED

The Disc Dump program uses the following peripherals.

- a disc unit on which the program file to be printed is located
- a disc unit for the SD, if the program file to be printed does not contain a •DDP program
- a printer.

3. PARAMETER CARDS

The Disc Dump program uses only an order card with the following format:

Col. 1 to 4	Order identification code: •DDP
Col. 5 & 6	Logical disc unit number (punched in column 6) of the unit on which the volume containing the file of programs to be printed is installed; if this is the System Disc, punch SD.

Col. 7 = 0 if only cylinder 0 is to be dumped

= ƀ if the entire program file is to be dumped including cylinder 0

= 1 if only type 1 program are to be dumped.

In this case, if columns 8 to 15 are not punched, printing begins with the first type 1 program in the program file index; otherwise, the program specified in columns 8 to 15 is dumped first. If columns 17 to 24 are blank, dumping ends with the last type 1 program; otherwise dumping ends with the program specified in columns 17 to 24. In this case, columns 16 to 25 should be blank.

2 when only the subroutine library is to be dumped.

In this case, if columns 8 to 13 are blank, dumping begins with the first subroutine specified in the subroutine library; otherwise it begins with the subroutine specified in columns 8 to 13. If the area between columns 14 and 19 is blank, the library is dumped completely; otherwise dumping ends with the subroutine specified in this area.

= 3 if only type 3 programs are to be dumped.

In this case, the same rules apply to punches in columns 8 to 15 and 17 to 24 as stated for type 1 programs; columns 16 to 25 are left blank if columns 8 to 15 or 17 to 24 are blank. If these areas contain the name of a program, punch "c" in columns 16 or 25 for a debugging program, or leave blank for a regular program.

Col. 26 = ƀ if each program is to be dumped completely

= 1 if only the indications in the program file index are to be dumped for each program requested.

Col. 27 to 72	Not used
Col. 73 to 80	Serial code (if desired).

4. DESCRIPTION AND EXECUTION

The program is made up of a single segment located at memory address 2048 (decimal).

It must be remembered that an order card can only be used to dump a single program file.

5. CHECKS AND ERROR MESSAGES

The program checks the operation of disc units and the validity of parameters punched in the order card. When an error is detected, it is indicated by means of one of the following messages:

0101 * D. UNIT N PERSISTENT ERROR

Persistent read error on logical disc unit N. Program terminates feasibility condition: NO.

0102 * D. UNIT N OUT OF ORDER

Logical disc unit N signalled out of order.
The program causes the printer to move the paper to the next page and halts at location 01DA. Pressing RUN retries the operation which caused the signal. If the signal repeats, the program terminates; feasibility condition: NO.

0103 *D. UNIT N PERSISTENT ERROR

Positioning error on logical disc unit N.
Program terminates; feasibility condition: NO.

0104 x NO FILE PROGRAM ON N D. UNIT

There is no program file in the volume on logical disc unit N.
Program terminates; feasibility condition: NO.

0105 x COLUMN NN OF ORDER CARD WRONG

The order card area beginning at column nn was incorrectly punched.
Program terminates; feasibility condition: NO.

0106 x TN . . . NC NOT FOUND

No type T program named N . . N with debugging code C in the program file index.
Program terminates; feasibility condition: NO.

0107 x UNIT N IS NOT A SD

Request for dumping of subroutine library from a volume on logical disc unit N which has no SD.

6. EXAMPLE

To dump all type 3 programs, beginning with the volume label and the program file label, of the volume installed on logical unit 2, starting with the first program file index and continuing through the program named FUSIONE 1, which is not a debugging program, the order card, which is linked to the others by the serial code ALFA, should have the following format:

Col. 1 to 4	•DDP
Col. 5 to 6	b̸2
Col. 7	3
Col. 8 to 15	b̸ . . . . b̸
Col. 16	b̸

Col. 17 to 24	FUSIONE 1
Col. 25	b̸
Col. 26	b̸
Col. 27 to 72	b̸ . . . b̸
Col. 73 to 76	ALFA
Col. 77 to 80	b̸ . . . b̸

•CMT COMMENT

1. FUNCTION

To print a comment line.

2. ORDER CARD

The order card has the following format:

Col. 1 to 4	Identification code: •CMT
Col. 5 to 72	Comment
Col. 73 to 80	Serial code (if desired)

3. EXECUTION

The comment is printed and Pilot 0 is initiated.

•HLT HALT

1. FUNCTION

To halt processing.

2. ORDER CARD

The order card has the following format:

Col. 1 to 4	Identification code: •HLT
Col. 5 to 72	Not used
Col. 73 to 80	Serial code.

3. EXECUTION

On a HALT instruction, Pilot 1 stops at memory address 01D6. Pressing RUN restarts Pilot 0.

•SDG SYSTEM DISC GENERATE

1. FUNCTION

This program personalizes a System Disc, that is it adapts it to a determined configuration. The order •SDG will apply to a volume already containing a System Disc.

The orders that must be carried out for the generation of a System Disc usually succeed each other in the following sequence:

•IND = assigns a serial number to a volume on which a System Disc will be recorded

•FLW = file label write

•SPD = duplication of a System Disc from the source volume onto the volume to generate

•SDG = adaptation of the new System Disc to the desired configuration.

2. SYSTEM UNITS REQUIRED

The program requires the following peripheral unit:

- a disc unit containing the source System Disc
- a disc unit containing the System Disc to be generated
- printer
- card reader

3. PARAMETER CARDS

The program requires one order card and several detail cards.

3.1. ORDER CARD

It has the following format:

Col. 1 to 4	Identification code •SDG
Col. 5 to 6	logical number (punched in column 6) of the disc unit which contains the System Disc to be generated or WD if the logical number has been punched in column 17 of the last •INT card read.
Col. 7 to 12	Serial number of the volume mounted on the working unit (b̸ if check is not desired)
Col. 13 to 72	not used
Col. 73 to 80	serial code if desired.

3.2. DETAIL CARDS

These cards specify the configuration desired. There must be a minimum of five and a maximum of seven such cards.

The following five cards must always be present :

- RDC - reader compile
- PRC - printer compile
- DSC - disc compile
- CDC - code compile
- MCC - memory capacity compile

The other two are required when the relative peripherals are present:

- TPC - tape compile
- PNC - punch compile

3.2.1. RDC card

It specifies the type of the card reader used in the configuration and the connector number assigned to it.

It has the following format:

Col. 1 to 4	identification code b̸RDC
Col. 5 to 6	reader type: - b̸1 = CRZ100, CRZ120 - b̸2 = to be assigned - b̸3 = to be assigned - b̸4 = CRP100
Col. 7 to 9	reader connection number (decimal) or blank if the reader has an integrated controller
Col. 10 to 80	not used.

3.2.2. PRC card

It specifies the type of the printer and the connection number assigned to it.
It has the following format:

Col. 1 to 4	Identification code b̸PRC
Col. 5 to 6	printer type: - b̸1 = Integrated printer (PRT 100, 110, 143, 144, 146, 163, 166). - b̸2 = Buffered printer (PRT 118, 120, 125, 130, 145, 147, 148, 149, 168, 169).

- b̸3 = Serial printer (PRT 105).
- b̸4 = to be assigned
- b̸5 = to be assigned

Col. 7 to 9	printer connector number (decimal) or blank if the printer has an integrated controller.
Col. 10 to 80	Not used.

3.2.3. DSC card

It specifies the type of the disc controller connected to the system, the number of the disc units involved and their connection numbers.

It has the following format:

Col. 1 to 4	Identification code b̸DSC
Col. 5 to 6	disc controller type: - b̸1 = DSC130 - b̸2 = to be assigned - b̸3 = to be assigned
Col. 7 to 8	number decimal disc units connected, from 1 to 10
Col. 9 to 11	connection number (decimal) of the first disc unit connected
Col. 12 to 14 Col. 36 to 38	connection number (decimal) of other possible disc units
Col. 39 to 80	Not used.

3.2.4. TPC card

It specifies the type of tape controller connected to the system and the number of tape units involved.

It has the following format :

Col. 1 to 4	identification code ƀTPC
Col. 5 to 6	tape controller type: - ƀ1 = GUN STANDARD 7 - ƀ2 = GUN STANDARD 7/9 - ƀ3 = to be assigned - ƀ4 = to be assigned - ƀ5 = to be assigned - ƀ6 = to be assigned
Col. 7	number (decimal) of tape units connected, from 1 to 8
Col. 8 to 80	not used.

If there are no tape units, the TPC card is not included.

3.2.5. PNC card

It specifies the type of punch used in the configuration and the connection number assigned to it.

It has the following format:

Col. 1 to 4	identification code ƀPNC
Col. 5 to 6	punch type: - ƀ1 = CRP100 - ƀ2 = CPZ101 - ƀ3 = CPZ103

Col. 7 to 9	punch connection number (decimal)
Col. 10 to 80	not used

If there is no punch in the system, the PNC card is not included.

3.2.6. CDC card

It specifies the punch code used in the configuration.

It has the following format:

Col. 1 to 4	identification code ƀCDC
Col. 5	punch code: - H = code Hollerith - B = code Bull
Col. 6 to 80	not used.

3.2.7. MCC card

It specifies the memory maximum capacity provided for in the configuration.
It has the following format:

Col. 1 to 4	identification code ƀMCC
Col. 5 to 6	memory capacity: - 08 = 8,192 location memory - 12 = 12,288 location memory - 16 = 16,384 location memory
Col. 7 to 80	not used

3.2.8. END card for end of order

This card must follow the detail cards:

Col.	1 to 4	identification code ƀEND
Col.	5 to 80	not used

4. DESCRIPTION AND EXECUTION

The program with decimal location address 2048 is made up of 57 segments. The first two segments make up the actual program and the other 55 are each subroutines; some of the subroutines will be inserted by the program cylinder zero of the System Disc to be generated, according to the current configuration.

The 55 segments can be divided into three groups; first group for disc subroutines, second group for tape subroutines and third group for read and print subroutines.

DISC SUBROUTINES

segment 3 = DSC130

segment 4 = to be assigned

segment 5 = to be assigned

TAPE SUBROUTINES

segment 6 = GUN STANDARD 7
segment 7 = GUN STANDARD 7/9
segment 8 = to be assigned
segment 9 = to be assigned
segment 10 = to be assigned
segment 11 = to be assigned
} when configuration includes a Hollerith code card reader.

segment 12 - 17 = as segment 6 - 11 when configuration including a Bull code card reader.

READ AND PRINT SUBROUTINES

segment 18 = integrated reader (CRZ100, CRZ120) integrated printer
segment 19 = integrated reader (CRZ100, CRZ120) buffered printer
segment 20 = integrated reader (CRZ100, CRZ120) serial printer
segment 21 - 32 = to be assigned
segment 33 = reader punch (CRP100) integrated printer
segment 34 = reader punch (CRP100) buffered printer
segment 35 = reader punch (CRP100) serial printer
segment 36 - 36 = to be assigned
} when configuration includes a Hollerith code card reader

With a card reader (Bull code), segment 38 - 57 corresponding to segments 18 - 37.

It should be noted that the only difference between the magnetic tape subroutines for a configuration using a Hollerith reader and the magnetic tape subroutines using a Bull reader lies in their location address.

4002 x D. UNIT N OUT OF ORDER

Unit number N indicates "OUT OF ORDER". The program makes a page change on the printer and halts at the address 01DA.
By depressing RUN the operation which caused the message is restarted; if that message reappears the program stops and the feasibility condition is NO.

4003 x D. UNIT N PERSISTENT ERROR

Error in positioning on unit number N. The program stops and the feasibility condition is NO.

4010 x NUMBER OF DISC UNIT WRONG

The disc unit logical number specified in the •SDG card is wrong; the program stops and the feasibility condition is NO.

4011 x NUMBER OF DETAIL CARDS WRONG

Some of the detail cards which must be present (RDC, PRC, DSC, CDC, MCC) have not been read.
The program stops and the feasibility condition is NO.

4012 x DISC UNIT NOT AVAILABLE

The serial number punched on the •SDG control card does not belong to the volume mounted on the working unit.
The program stops and the feasibility condition is NO.

4013 x DETAIL CARD WRONG

An error has been detected in a detail card. The program stops and the feasibility condition is NO.

7. EXAMPLE

A System Disc (mounted on logical unit 1) is to be adapted to a configuration provided with an 8,192 location memory and connected to a CRP100 that is used as a card reader punch (Hollerith code, a PRINTER300 and a disc subsystem DSS130.

The following cards must be punched:

1. Order Card

Col. 1 to 4	●SDG
Col. 5 to 6	b̸1
Col. 7 to 80	not used.

2. RDC detail card

Col. 1 to 4	b̸RDC
Col. 5 to 6	b̸4
Col. 7 to 9	064
Col. 10 to 80	not used.

3. PRC detail card

Col. 1 to 4	b̸PRC
Col. 5 to 6	b̸1
Col. 7 to 80	not used.

4. DSC detail card

Col. 1 to 4	b̸DSC
Col. 5 to 6	b̸1
Col. 7 to 8	02
Col. 9 to 11	032
Col. 12 to 14	033
Col. 15 to 80	not used.

5. PNC detail card

Col. 1 to 4	b̸PNC
Col. 5 to 6	b̸1
Col. 7 to 9	064
Col. 10 to 80	not used.

6. CDC detail card

Col. 1 to 4	b̸CDC
Col. 5	H
Col. 6 to 80	not used.

7. MCC detail card

Col. 1 to 4	b̸MCC
Col. 5 to 6	08
Col. 7 to 80	not used.

8. END card for end of order

Col. 1 to 4	b̸END
Col. 5 to 80	Not used

•RST RESTART

1. FUNCTION

The •RST order resumes program execution from a memorized check-point. The check-point is written into any standard file by means of the macro-instruction CHKP in the case of a tape or by means of the macro-instruction DCKP into a previously specified volume area.

Before resuming program execution, the •RST order fills the memory and resets the I/O files on tape and on disc.

2. SYSTEM UNITS REQUIRED

The program requires the following peripherals for execution:

- a disc unit that contains the System Disc or the Master Disc
- as many disc units or tape units as are required by the program concerned
- a card reader
- a printer.

3. PARAMETER CARDS

Two types of order cards are used depending on whether the check-point has been written on tape or on disc.

Using the detail cards that specify the configuration desired the •SDG program modifies some of the source program areas contained in cylinder zero of the System Disc to be generated, is :

a) it compiles three branches to the subroutines, that is two branches to the tape subroutines (addresses 0068 and 0074) and one branch to the read and print routine (address 006C) according to the punch code used in the configuration.

b) it writes into the source program second segment some bits that specify the new configuration.

c) in segments 2, 3, 5 and 6 of the Source Program, it replaces the read and print subroutines the tape and disc subroutines by those that area indicated in the detail cards, via the 55 segments forming the program.

d) it writes the new index of the cylinder zero into sectors 21 and 22. This index is constituted by records of 7 positions each; it ends with a character FF for end of table. The first two characters in each record indicate the length (binary) of the data contained on the cylinder; the next two characters represent the location address (binary) of this data. The last three characters stand for the address (decimal) of the sector in which the data is written.

5. CHECKS AND ERROR MESSAGES

After checks have been performed on the peripherals and the control cards, the program indicates the errors by means of the following messages:

4001 x D. UNIT N PERSISTENT ERROR

Persistent error in reading, writing or checking on unit number N. The program stops and the feasibility condition is NO.

3.1. ORDER CARD FOR A DISC-ORIENTED

It has the following format:

Col. 1 to 4	identification code •RST
Col. 5 to 12	program name
Col. 13 to 18	six blank characters
Col. 19 to 72	not used
Col. 73 to 80	serial code, if any.

3.2. ORDER CARD FOR A TAPE-ORIENTED CHECKPOINT

It has the following format:

Col. 1 to 4	identification code •RST
Col. 5 to 80	number of the checkpoint from which the program is to be restarted
Col. 9	logical number of the tape unit containing the tape on which the checkpoint has been written
Col. 10	code 0 for a tape with even parity check code P if it is written with odd parity check
Col. 11	code indicating the packing density; it may have the following values: L = 800 bpi M = 556 bpi N = 200 bpi
Col. 12 to 17	serial number of the tape containing the checkpoint.

Col. 18	tape-recording mode according to the following values: 7 - for seven track tape 9 - for nine track tape
Col. 19 to 72	not used
Col. 73 to 80	serial code, if any.

4. DESCRIPTION AND EXECUTION

The program is made up of four segments; its location address is 2300 (decimal).

THE FIRST segment carries out the following functions:

A. Checks the parameters contained in the order card.

B. Searches for the checkpoint written on disc or tape, according to the order card.

C. Loads, from tape or disc into memory, a series of data including the file table needed to carry out the checks on the files.

D. Loads the second segment into memory.

THE SECOND SEGMENT performs the following functions:

A. Checks and resets the peripherals according to the data contained by the file table. This is done only for the standard files that were already open when the checkpoint was written.

A. Loads into memory the third segment if the checkpoint has been written on tape or the fourth segment if the checkpoint has been written on disc.

THE THIRD SEGMENT carries out the following functions:

A. Fills the memory from the program location address. It also fills the reserved areas concerning the program including the print and read areas.

B. May call a user sequence, to check non standard files or to perform any other operation before the program is restarted.

C. Initiates the program from the instruction that follows the macro-instruction which generated the checkpoint.

The part of program which carries out these functions fills the read area and the print areas.

The fourth segment has the same functions as the third one but for a disc oriented checkpoint instead of tape.

It should be noted that the system configuration at the time when the program is restarted must be the same as when the checkpoint was written: the same logical numbers must be assigned to the disc or tape units in order to make sure that the program to be restarted will use the same volumes and the same peripherals.

5. ERROR CHECKS AND MESSAGES

The program performs a number of checks on the working of the units concerned and the validity of the parameters punched on the order card.

The errors detected are indicated by means of the following messages:

4501 x D. UNIT N PERSISTENT ERROR

Persistent error in reading on the disc unit with logical number N. The program stops and the feasibility condition is NO.

4502 x D. UNIT N OUT OF ORDER

The disc unit with logical number N is OUT OF ORDER.
The program makes a page change on the printer and halts at the address 01DA.
By depressing RUN the operation which caused the message is restarted, if the message reappears, the program stops and the feasibility condition is NO.

4503 x D. UNIT N PERSISTENT ERROR

Persistent error in positioning on the disc unit with logical number N. The program stops and the feasibility condition is NO.

4504 x UNIT N NO STANDARD FILE

Tape unit N contains a non standard file.
Program stops; feasibility condition: NO.

4505 x UNIT N PERSISTENT ERROR

Persistent error in reading or writing on the tape unit with logical number N. Program stops; feasibility condition: NO.

4506 x UNIT N OUT OF ORDER

Tape unit N is OUT OF ORDER.
If the out of order condition occurs after a rewind, the program halts at the address 01DA. If that condition occurs after a read or write instruction, the program stops at the address 02A6 and the feasibility condition is NO.

4507 x UNIT N PERSISTENT ERROR

Error in positioning on tape unit N.
Program stops at the address 02A6; feasibility condition: NO.

4508 x ORDER CARD COLUMN NN ERROR

Error in one of the columns 9, 10, 11 or 18 of an order card used for a tape-oriented restart or columns 13-18 for a disc-oriented restart.
The program stops and the feasibility condition is NO.

4509 x CHKP-FILE NOT FOUND

The System Disc contains no CHKP file. Program stops; feasibility condition : NO.

4510 x CHKP-FILE NOT AVAILABLE

The general CHKP file cannot be used. In case of special CHKP file, the error may result of a difference between the actual name of the file and the one indicated in the order card.
The program stops and the feasibility condition is NO.

4511 x CHKP BLOCKS ARE INCORRECT

The blocks written during the checkpoint do not exactly correspond to the part of the memory to be filled again.
The program stops and the feasibility condition is NO.

4512 x D. UNIT N VOLUME M ... M ABSENT

The serial;number of the volume contained in the file table (M... M) is not the same as the number of the volume mounted on unit N.
The program stops and the feasibility condition is NO.

4514 x UNIT N : VOLUME M ... M ABSENT

The volume indicated in search for the checkpoint, and which is mounted on tape unit N, has a number other than the one punched in the order card. The program stops and the feasibility condition is NO.

4515 x UNIT N CHKP NNNN ABSENT

The checkpoint required is not found on the tape indicated.
The program stops and the feasibility condition is NO.

4516 x D. UNIT N NOT AVAILABLE VOLUME

The disc unit N does not contain a volume label or a file label corresponding to the labels found in the file table. A print out of the volume file label found in volume N is carried out before the error message.
The program stops and the feasibility condition is NO.

4599 x END RESTART DISC AS CHKP NNNN

End of the restart program.

6. EXAMPLE

6.1. TAPE-ORIENTED CHECKPOINT

A program is to be restarted from checkpoint number 0033.
The checkpoint has been written on the tape contained in tape unit 3 and described as follows:

- 9 tracks
- density 800 bpi
- odd parity check
- serial number 0A7913.

The card to be punched must have the following format:

Col. 1 to 4	•RST
Col. 5 to 8	0033
Col. 9	3
Col. 10	P
Col. 11	L
Col. 12 to 17	0A7913
Col. 18	9
Col. 19 to 80	not punched.

6.2. DISC-ORIENTED CHECKPOINT

The program PROVA is to be restarted from the last checkpoint written.

The card to be punched must have the following format:

Col. 1 to 4	•RST
Col. 5 to 9	PROVA
Col. 10 to 80	not punched.

●SGE SAMPLES GENERATE

1. FUNCTIONS

The ●SGE program writes a sample file on one of the tapes of the system data are written on detail cards. Resulting files may be standard or non-standard, and blocks may or may not be numbered. If the file is to be standard with numbered tapes it is automatically closed by the program.

If records with variable length are generated, writing of record lengths is left to the user.

2. SYSTEM UNITS REQUIRED

The program requires the following peripherals:

- a disc unit containing the System Disc
- a printer
- a card reader
- a tape unit.

3. PARAMETER CARDS

3.1. ORDER CARD

It has the following format:

Col. 1 to 4	identification code •SGE
Col. 5 to 6	code indicating the logical number of the tape unit on which writing takes place; it can assume values ranging from ƀ1 to ƀ8.
Col. 7	code indicating a even or odd parity checking: P = odd parity check O = even parity check
Col. 8	code indicating the packing density; it can assume the following values: L = 800 bpi M = 556 bpi N = 200 bpi
Col. 9	code indicating the format of data punched on detail cards; it can assume the following values: X = the detail cards contain hexadecimal data that can be accompanied by alphanumeric characters preceded by a character . (period). In this case, for every two characters punched on a card only one characters is written on tape. This format which enables writing every possible configuration of the code GECD, can only be used with 9 track tapes. ƀ = the detail cards contain alphanumeric data. For each character punched on a card a character is written on tape.

Col. 10 to 14	Number of blocks to be reserved on the tape. If a standard file is desired and the tape is labelled, three blocks must be reserved.
Col. 15 to 72	not used
Col. 73 to 80	Serial code, if any.

3.2. DETAIL CARDS

The detail cards contain the data to be written into the file. There are four types of detail card:

1. Cards BLK containing the characters to be written
2. Cards SBL for start of block
3. Cards EBL for end of blocks
4. Cards TPM for tape mark write.

The detail card input sequence is as follows:

SBL - BLK - BLK ...... BLK - EBL - SBL - BLK - BLK .....

All detail cards must be immediately followed by the card END.

Each of these detail cards is described below:

Card SBL

It must be placed before all the BLK cards that make up one block.

Col. 1 to 4	identification code ƀSBL
Col. 5 to 9	number of the block to be written. Not punched if no number is assigned to the block
Col. 10 to 80	not used.

Card BLK

Col.	1 to 4	identification code ƀBLK
Col.	5 to 6	Number of characters punched on the card and to be written on tape. This number may not be greater than 74 since columns 7 - 80 are reserved for the characters to be written.
Col.	7 to 80	characters to be written.

Card EBL

It must come after all the BLK cards that make up one block.

Col.	1 to 4	identification code ƀEBL
Col.	5 to 80	not used.

Card TPM

Writes a tape mark.

Col.	1 to 4	identification code ƀTPM
Col.	5 to 80	not used.

3.3. END CARD FOR END OF ORDER

This card must come immediately after all the detail cards. It has the following format:

Col.	1 to 4	identification. code ƀEND
Col.	5	code indicating the type of file closing: V = the file is closed by label EOV

F = the file is closed by writing of label EOF

b̸ = the file has already been closed by the detail cards. The program merely rewinds the tape. The program may directly close the file by writing the label EOV or EOF only when the tape has a standard label and the order card requires the reservation of two blocks or more. Otherwise the file must be closed by means of the detail cards and the program checks the presence of code b̸ in column 5.

Col. 6 to 80 not used.

4. DESCRIPTION AND EXECUTION

The program is made up of one segment and located at address 2300 (decimal). It first checks the parameters contained in the order card; if no errors are detected it writes the file. In doind so it sets the tape to be written after the blocks to be reserved. When no block is to be reserved, the program rewinds the tape and carries out two erasures.

Then the program reads, checks and prints out the detail cards while making up and writing the blocks.

Once the END card has been read the program closes the file as required by the END card and rewinds the work tape.

5. CHECKS AND ERROR MESSAGES

The program checks the order card for its parameters.

- Code indicating the tape unit on which writing takes place (col. 5 - 6) The program checks that one of the accepted values has been punched

- Code indicating even or odd parity check (col. 7).
  The program checks that one of the two values O or P has been punched.

- Code indicating the packing density (col. 8).
  The program checks that it has one of the accepted values.

- Code indicating the format of the data contained in the detail cards (col. 9). The program checks that it has value X or b̸.

The program checks the identification codes and the read sequence:
(SBL - BLK - EBL) of the detail cards.

In addition the program checks the card BLK for the number of characters to be written (col. 5 - 6) that is not greater than 74. The program also checks that column 5 of the END card contains one of the codes V, F, or b̸ indicating to reserve two blocks or more.
Otherwise the program checks that the second block is the beginning file label (HDR).

Errors are indicated by means of the following messages:

2301 x TAPE UNIT CODE ERROR

The code indicating the tape to be written is wrong.
Program stops at address 02A6; feasibility condition: NO.

2302 x CHECK CODE ERROR

The code indicating the type of check (even or odd parity check) is not correct. Program stops at address 02A6; feasibility condition: NO.

2303 x DENSITY CODE ERROR

The code indicating the packing density is not correct. The program stops at the address 02A6 and the feasibility condition is NO.

2304 x READING TYPE CODE ERROR

The code indicating the format of the data punched on the detail cards is incorrect.
The program stops at the address 02A6 and the feasibility condition is NO.

2305 x LENGTH ERROR

A BLK card indicates a greater number of characters to be written than 74.
The program stops at the address 02A6 and the feasibility condition is NO.

2306 x LAST LABEL CODE ERROR

In the END card the code indicating the type of close is wrong.
The program stops at the address 02A6 and the feasibility condition is NO.

2308 x UNIT X : OUT OF ORDER

The tape unit is OUT OF ORDER.
If the out of order condition occurs after a rewind control the program halts at the address 01DA. The operator can put the tape unit back in order and depress RUN.
If this condition occurs after a read or write instruction the program stops at the address 02A6 and the feasibility condition is NO.
When it occurs during final rewinding, the program terminates normally.

2309 x UNIT X : PERSISTENT ERROR

Persistent error in reading or writing on the tape unit indicated.
The program stops at the address 02A6 and the feasibility condition is NO.

2310 x SEQUENCE ERROR IN SUB-ORDER CARDS

The detail cards are out of sequence. The program stops at the address 02A6; feasibility condition: NO.

2311 x UNIT X · END OF TAPE

End of tape on tape unit X. The program rewinds the tape and stops at the address 02A6.
The feasibility condition is NO.

2312 x BLOCK LENGTH ERROR

A block has more than 3584 characters for an 8 K memory or more than 4096 characters for a 12 K or 16 K memory.
The program stops at the address 02A6.
The feasibility condition is NO.

2313 x HDR LABEL NOT FOUND

Label HDR missing on the tape.
The program continues normally. It stops at the address 02A6 if the card END is not ƀ in column 5.
The feasibility condition is NO.

2307 x END OF SAMPLES GENERATE

End of program.

6. EXAMPLE

A sample file is to be written on tape 1 with packing density 800 bpi and odd parity check; it is a nine-track tape. The data on cards are punched in hexadecimal and the file is standard with numbered blocks.

Following are descriptions of the order card, the detail cards SBL and EBL concerning the first block, and the END card as the file is closed by label EOF.

Order Card

Col. 1 to 4	●SGE
Col. 5 to 6	b̸1
Col. 7	P
Col. 8	L
Col. 9	X
Col. 10 to 14	00003
Col. 15 to 80	not punched.

Detail card SBL

Col. 1 to 4	b̸SBL
Col. 5 to 9	00001
Col. 10 to 80	not punched

Detail card EBL

Col. 1 to 4	b̸EBL
Col. 5 to 80	not punched

END card

Col. 1 to 4	b̸END
Col. 5	F
Col. 6 to 80	not punched.

•TDP TAPE DUMP

1. FUNCTION

This program prints out tape contents at given intervals.
Printing may be in alphanumeric characters or in hexadecimal.
Blocks cannot have more than 4096 characters.

2. SYSTEM UNITS REQUIRED

The program requires the following peripherals:

- a disc unit containing the System Disc
- a tape unit containing the tape to be dumped
- a printer.

3. PARAMETER CARDS

Only an order card is required; it has the following format :

Col. 1 to 4	identification code •TDP
Col. 5 to 6	logical number (punched in column 6) of the tape unit containing the tape to be dumped
Col. 7	Code P indicating an odd parity check Code O indicating an even parity check

Col. 8	Code indicating the packing density. L = 800 bpi M = 556 bpi N = 200 bpi (x)
Col. 9	code indicating the type of print out. X = hexadecimal b̸ = alphanumeric
Col. 10 to 14	sequence number of the first block to be printed out. In the absence of the number the print out starts from the first block written on the tape, that is from the label INT1 or VOL1 in the case of standard tapes.
Col. 15 to 19	number of blocks to be printed out. In the absence of this number the print out ends with label EOV or EOF.
Col. 20 to 72	Not used.
Col. 73 to 80	Serial code, if any.

4. DESCRIPTION AND EXECUTION

The program is made up of one segment with location address 2300 (decimal). It first checks the order card for its parameters. Though an incorrect parameter might be found the check is still completed in order to indicate all the errors detected.

(x) Using GUN 7/9 and 9 track tapes, only code L is available.

Using GUN 7/9 and 7 track tapes, codes L, M, and N are available.

Using GUN 7 codes M, N and L are available. In this case code L = 556 bpi.

If all parameters are correct the tape to be dumped is rewound and set before the first block to print out.

Program stops after the number of blocks indicated in the order card or after the end label has been printed. Then the tape is rewound.

In the print out the following line comes first:

```
1901 x TAPE DUMP OF UNIT X FROM BLOCK NNNNN/BLOCKS :
        XXXXX
```

with X standing for the name of the tape unit containing the tape to be dumped, NNNNN for the first block number and XXXXX for the number of blocks to print out.

Prior to each block, a line is printed that contains the corrent block number and the number of characters in the block.

If the last block number is not specified, the program stops after the print out of the end label EOV1 or EOF1.

5. ERROR CHECKS AND MESSAGES

The program performs formal checks on the codes punched on the order card (columns 5 – 9).

The following messages may be indicated:

```
1901 x TAPE DUMP OF UNIT X FROM BLOCK NNNNN/BLOCKS XXXXX
```

Print out heading

```
1902 x END OF TAPE DUMP
```

End of program

1903 x UNIT X PERSISTENT ERROR

Persistent error in reading on tape.
The tape is rewound and control is transferred to Pilot 0.
Feasibility condition: YES.

1904 x UNIT X : OUT OF ORDER

The tape unit is out of order.
If the out of order condition occurs after a rewind control, the program halts at the address 01DA. The operator can put the tape unit back in order and depress RUN.
If this condition occurs after a read instruction, the program stops at the address 02A6 and the feasibility condition is NO.
When it occurs during the final rewinding the program terminates normally.

The messages that follow (1905 - 1908) concern errors detected in the order card. Each time the program will complete the reading of the card. Then it stops at the address 02A6 and the feasibility condition is NO.

1905 x TAPE UNIT CODE ERROR

The code (columns 5 - 6) indicating the logical number of the tape unit containing the tape to be dumped is not correct.

1906 x CHECK CODE ERROR

The code indicating an evenoor odd parity check (col. 7) is neither O nor P.

1907 x DENSITY CODE ERROR

The code indicating the packing density (col. 8) is unknown

1908 x DUMPING-TYPE CODE ERROR

The code indicating the desired print out type (col. 9) is neither b̸ nor X.

6. EXAMPLE

It is desired to print out, in hexadecimal, the tape contained on tape unit 3 which has been written with a packing density of 800 bpi.

The following control card must be punched:

Col. 1 to 4	•TDP
Col. 5 to 6	03
Col. 7	P
Col. 8	L
Col. 9	X
Col. 10 to 80	not punched.

•TRP TAPE RECORD BY RECORD PRINT

1. FUNCTION

This program prints the content of a data tape record by record. The records may be of fixed or variable length format. Records of variable length must contain the "record length" field. Tapes must have a standard organization: 2 header labels followed by 1 tape mark,block numbers and a trailer label. The listing may be in alphanumeric or hexadecimal. It is not necessary to print the whole file. Certain areas may be selected by means of parameters.

2. SYSTEM UNITS REQUIRED

This order requires the following peripherals:

- a printer
- a disc unit for the System Disc
- a tape unit for the tape to be printed

3. PARAMETER CARDS

This Order uses one Order card only whose format is indicated below:

Col. 1 to 4	•TRP
Col. 5 to 6	name of the unit on which the tape to be printed, is loaded. – W1 = Work Tape 1 – W2 = Work Tape 2 – W3 = Work Tape 3

otherwise, the select number of the unit can be directly indicated in col. 6

Col. 7	– P for even parity. – O for odd parity.
Col. 8	code specifying the recording density K = 1600 bpi L = 800 bpi M = 556 bpi N = 200 bpi
Col. 9	code specifying the printing mode X = hexadecimal print ƀ = alphanumeric print
Col. 10 to 14	sequence number of the first block to be printed, with leading zeros. If not indicated, printing starts from the first block of the tape, that is from VOL1 label for standard tapes.
Col. 15 to 19	number of the blocks to be printed, with leading zeros. If not indicated, printing ends with either end of volume or end of file.
Col. 20 to 23	record length if the record is FL or UN, with leading zeros. ƀƀƀƀ if the record is VL.
Col. 24 to 72	not used
Col. 73 to 76	optional ID code
Col. 77 to 80	not used.

4. DESCRIPTION AND EXECUTION

The program proceeds as follows:

First, the order card is checked for any incorrect parameters.

If parameters are correct, the tape to be printed is rewound and positioned in front of the first block to be printed.

The program ends after printing either the number of blocks indicated in the order card, or the trailer label. At the end, the tape is rewound.

If the number of the last block to be printed is not specified, the program ends by printing the trailer label EOV1 or EOF1.

If the block number is specified in cols. 10 - 14, this number must be increased by 3 to account for 2 header labels and 1 TM.

If a record is longer than one print line, printing will continue on the following line.

In the case of hexadecimal print, the corresponding alphanumeric values will appear on the following line.

5. CHECKS AND ERROR MESSAGES

This order has no standard messages (no message number is indicated).

**** PERSISTENT ERROR

Read persistent error. By depressing START the operation is attempted five more times. Program execution stops at location 01C8; feasibility condition: NO.

*MTU N. X OUT OF ORDER

Logical magnetic tape unit X out of order. Program execution stops at location 02A6; feasibility condition: NO.

* PARAMETER ERROR: CORRECT AND BEGIN

Parameter card is wrong. Character ↑(arrow) shows the first position of the wrong field;correct the card and restart from the beginning.

* END OF DUMP

End of program.

* END OF FILE

The trailer label has been detected.

6. EXAMPLE

Print in hexadecimal the contents of the tape loaded on unit 4, written at 800 bpi with records of fixed length of 107 chrts. each. Start printing from block 10.
The following cards must be punched.

•TLW TAPE LABEL WRITE

1. OBJECT

This program writes beginning volume, end of volume, beginning file and end of file standard labels on any number of tapes.

2. SYSTEM UNITS REQUIRED

The program requires the following peripherals:

- a disc unit containing the System Disc
- a reader
- a printer
- a number of tape units equal to the number of tapes that are to be labelled.

3. PARAMETER CARDS

3.1. ORDER CARD

It has the following format:

Col. 1 to 4	identification code •TLW
Col. 5 to 72	not used
Col. 73 to 80	serial code, if any.

3.2. DETAIL CARDS

Each detail card contains the parameters that concern a tape to be labelled. It has the following format:

Col. 1 to 4	identification code ƀTLW
Col. 5	code 7 for 7-track tape code 9 for 9-track tape
Col. 6 to 7	logical number (col. 7) of the tape unit containing the tape to be labelled.
Col. 8	- code P indicating an odd parity check - code O indicating an even parity check
Col. 9	code indicating the packing density: L = 800 bpi M = 556 bpi N = 200 bpi (x)
Col. 10	- code V if an end of volume label EOV1 is to be written - code F for an end of file EOF1
Col. 11 to 16	volume serial number
Col. 17 to 26	user's name
Col. 27 to 43	file name

(x) Using GUN 7/9 and 9-track tapes, only code L is available.

Using GUN 7/9 and 7-track tapes, codes L, M and N are available.

Using GUN 7, codes M, N, and L are available. In this case code L = 556 bpi.

Col. 44 to 49	serial number of the file first volume
Col. 50 to 53	volume sequence number in the file
Col. 54 to 59	Creation date expressed in the form ƀYYDDD. In the absence of that date, the current date loaded by the control record •INT is indicated. The cancellation date is always the current date.
Col. 60 to 80	Not used.

3.3. END CARD FOR END OF ORDER

That card must follow the detail cards:

Col. 1 to 4	ƀEND
Col. 5 to 80	Not used.

4. DESCRIPTION AND EXECUTION

The program is made up of one segment at location address 2300 (decimal) It first reads and checks the first detail card.

If the parameters are correct, the program passes on to tape number labelling: it rewinds the tape, carries out two successive erasures and then writes the labels and the tape marks.

Though an incorrect parameter may be found, the check is still completed in order to indicate all the errors detected.

If a detail card contains an error the corresponding tape is not labelled.

The program also reads the next detail card and writes the labels on the tape concerned, except when the identification code contained in the detail card has been detected as wrong (See message 1801).
The program terminates after reading of the END card.

Each tape then contains the following labels and tape marks:

- beginning volume label VOL1 or INT1 for 7-track tapes
- beginning file label HDR1
- two tape marks
- end of volume EOV1 or end of file label EOF1 according to the option found in the detail card
- one or two tape marks according to whether the preceding label is EOV1 or EOF1.

If incorrect detail cards have been found and the corresponding tapes have not been labelled the program terminates and the feasibility condition is NO.

The labelled tapes are rewound.

5. CHECKS AND ERROR MESSAGES

The program checks the detail cards for the parameters required in columns 5 - 10. It also checks that the date is expressed in numeric characters.

Errors are indicated by means of the following messages:

1801 * SUB-ORDER CARD NOT ACCEPTED

Detail cards wrong. The program has read a card that contains neither ƀTLW nor ƀEND in the first columns. The tape is not labelled. The program stops at the address 02A6 and the feasibility condition is NO.

1802 x TRACKS NUMBER ERROR

The track number is wrong (column 5 contains neither 7 nor 9).
The tape is not labelled and the feasibility condition is NO.

1803 x TAPE UNIT CODE ERROR

The code indicating the tape unit containing the tape to be labelled is unknown.
The tape is not labelled and the feasibility condition is NO.

1804 x CHECK CODE ERROR

The code indicating even parity check or odd check is incorrect (column 8 contains neither P nor O).
The tape is not labelled and the feasibility condition is NO.

1805 x DENSITY CODE ERROR

The code indicating the packing density is not correct (column 9 contains neither L, M, nor N).
The tape is not labelled and the feasibility condition is NO.

1806 x LAST LABEL CODE ERROR

The code indicating the type of the end label is neither V nor F in column 10.
The tape is not labelled and the feasibility condition is NO.

1807 x DATE CODE ERROR

The date is expressed in non-numeric characters.
The tape is not labelled and the feasibility condition is NO.

1808 x UNIT X : OUT OF ORDER

The tape unit is out of order. If the out of order condition occurs after a rewind control the program halts at the address 01DA.
The operator can put the tape unit back in order and depress RUN.

If this condition occurs after a read or write instruction the program stops at the address 02A6 and the feasibility condition is NO.
When it occurs during the final rewinding the program terminates normally.

1809 x UNIT X : PERSISTENT ERROR

Persistent error in writing.
The program stops at the address 02A6 and the feasibility condition is NO.

1810 x END OF TAPE LABEL WRITE

End of program.

EXAMPLE

Tapes 1 and 2 are to be labelled. Labels are to be written with packing density 800 bpi and odd parity check.

In both tapes the end of file label must be written.

The number labelling date is the current date.

The following cards must be punched:

Control card

Col. 1 to 4	●TLW
Col. 5 to 80	not punched.

First detail card TLW

Col. 1 to 4	ƀTLW
Col. 5	9
Col. 6 to 7	ƀ1
Col. 8	P
Col. 9	L
Col. 10	F
Col. 11 to 16	0A1251
Col. 17 to 26	OGE UDMS
Col. 27 to 43	SOFTWARE DISTR.
Col. 44 to 49	0A1251
Col. 50 to 53	0001
Col. 54 to 80	not used.

Second detail card TLW

Col. 1 to 4	ƀTLW
Col. 5	9
Col. 6 to 7	ƀ2
Col. 8	P
Col. 9	L
Col. 10	F
Col. 11 to 16	0A1205
Col. 17 to 26	OGE UDMS
Col. 27 to 43	SERVIZIO
Col. 44 to 49	0A1205

Col. 50 to 53	0001
Col. 54 to 80	not used.

END card

Col. 1 to 4	ƀEND
Col. 5 to 80	not punched.

•TDC TAPE DUPLICATE

1. PURPOSE

This program is designed to duplicate any tape, whether it contains standard or non-standard files or programs (System Tape, Library Tape or Master Tape).

2. SYSTEM UTILIZATION

The program uses the following peripheral units:

- Card reader
- Printer
- Tape unit on which tape to be duplicated is mounted
- Tape unit on which the tape to be written is mounted
- Tape unit for the System Tape.

3. PARAMETER CARDS

The program requires a single order card with the following format:

Col.	1 to 4	Identification code: •TDC
Col.	5 and 6	File type code of the tape to be duplicated: - b̸P if the tape to be duplicated contains programs (ST, LT or MT)

	- DS if the tape to be duplicated contains data with a standard format - DN if the tape to be duplicated contains data with a non-standard format.
Col. 7 and 8	Code identifying the tape to be duplicated. It may have one of the following values: - 01 through 08 to indicate the logical numbers of the tape units.
Col. 9	Code specifying the type of check for the tape to be duplicated: - P to indicate an odd parity check - O to indicate an even parity check.
Col. 10	Code specifying the recording density for the tape to be duplicated. It may have one of the following values(x): - L to indicate 800 bits per inch (bpi) - M to indicate 556 bits per inch (bpi) - N to indicate 200 bits per inch (bpi).
Col. 11 to 16	Serial number of the tape to be duplicated. If no check required, these columns are left blank.
Col. 17 and 18 Col. 19 Col. 20 Col. 21 to 26	As for columns 7 through 16, but for the tape to be written.

(x) Using a 7/9 track controller and 9-track tapes, only code L is accepted. Using a 7/9 track controller and 7-track tapes, codes L, M and N are accepted.. Using a 7 track controller, codes L, M and N are accepted. In this case code L = 556 bpi.

Col. 27	- Code C if the file contained in the tape to be duplicated must be recorded next to a file already existing on the tape to be written. - Otherwise, leave blank. It should be noted that code C can only be used if code DS is punched in columns 5 and 6.
Col. 28 to 44	Name of the file if this is to be changed on the tape to be written. Otherwise, leave blank. The name of the file can be changed on the tape to be written only if the file contains standard-format data and the file to be duplicated is not to be recorded next to a file already existing on the tape to be written. (Code DS in columns 5 and 6, and code b̸ in column 27).
Col. 45 to 50	Serial number of the file if this is to be changed on the tape to be written. Otherwise, leave blank. The restrictions are the same as for the file name (col. 28 through 44).
Col. 51 to 53	Number of tape retention days if the purge date is not to be the same as the writing date (day's date). Otherwise, leave blank. The dates in the HDR1 label are updated only if code DS is punched in columns 5 and 6 and code b̸ in column 27.
Col. 54 to 58	Number of blocks, starting from the first, which are not to be reproduced from the tape to be read. If the duplication is to be made starting from the first block, leave these columns blank.

Col. 59 - 63	number of blocks to be retained on the tape on which recording is effected. Nothing is punched if no block is to be retained. In this case the program effects two deletions. The data punched in columns 54 - 63 is considered in the case of non-standard tapes only (code DN in column 5 - 6)
Col. 64	- code N, if the blocks have not been initialized - blank, if they have been. This option is valid only if code C is punched in column 27
Col. 65	- code V, if the output tape is to be closed with the label EOV - any code, other than V, to close the tape with label EOF.
Col. 66 - 72	not used
Col. 73 - 80	serial code (optional).

4. PROGRAM DESCRIPTION AND EXECUTION

The program is made up of two segments. Initially, it checks the order card parameters and, if no error is detected, proceeds to duplicate the tape to be read. Since either the whole or only part of the tape may need to be duplicated, the program first positions the two tapes and then carries out the duplication as requested in the order card.
If the input tape contains check-points, these are not written.

5. CHECKS AND ERROR MESSAGES

The checks performed on the peripheral units and on the contents of the order card may result in the print out of one or more of the following messages:

2501 x UNIT X · PERSISTENT ERROR

Persistent reading or writing error on the unit indicated.
The program halts at address 02A6. Feasibility condition: NO.

2502 x UNIT X · OUT OF ORDER

Unit out of order. If the out-of-order condition occurs after a rewind order, the program stops at a HLT at address 01DA.
The operator may reset the unit and press START.
If this condition occurs after a read or write instruction, the program halts at address 02A6 and the feasibility condition is set to : NO.

2504 x FILE TYPE CODE ERROR

Error in the file type code (cols. 5 and 6). The program halts at address 02A6.
Feasibility condition: NO.

2505 x INPUT TAPE CODE ERROR

Error in the code identifying the tape to be duplicated (cols. 7 and 8).
The program halts at address 02A6.
Feasibility condition: NO.

2506 x INPUT TAPE CHECK CODE ERROR

The code indicating the type of check to be made on the tape to be duplicated is invalid (col. 9).
The program halts at address 02A6.
Feasibility condition: NO.

2507 x INPUT TAPE DENSITY CODE ERROR

The code indicating the recording density for the tape to be duplicated is invalid (col. 10).
The program halts at address 02A6.
Feasibility condition: NO.

2508 x INPUT TAPE SERIAL NUMBER XXXXXX NOT EXACT
YYYYYY EXACT

The serial number of the tape to be duplicated is incorrect (cols. 11 through 16).
The program halts at address 02A6.
Feasibility condition: NO.

2509 x OUTPUT TAPE CODE ERROR

The code identifying the tape to be duplicated is invalid (cols. 17 and 18.
The program halts at address 02A6.
Feasibility condition: NO.

2510 x OUTPUT TAPE CHECK CODE ERROR

The code indicating the type of checks to be made on the tape to be written is invalid (col. 19). The program halts at address 02A6.
Feasibility condition: NO.

2511 x OUTPUT CHECK TAPE DENSITY CODE ERROR

The code indicating the recording density for the tape to be written is invalid (col. 20).
The program halts at address 02A6.
Feasibility condition: NO.

2512 x OUTPUT TAPE SERIAL NUMBER XXXXXX NOT EXACT YYYYYY EXACT

The serial number of the tape to be written is incorrect (cols. 21 through 26).
The program halts at address 02A6.
Feasibility condition: NO.

2513 x COMPLEMENT CODE ERROR

The code punched in column 27 is invalid.
The program halts at address 02A6.
Feasibility condition: NO.

2514 x RETENTION PERIOD ERROR

The code specifying the retention period is invalid (cols. 46 through 48).
The program halts at address 02A6.
Feasibility condition: NO.

2515 x INPUT TAPE BLOCK NUMBER ERROR

The code indicating the number of blocks that are not to be written is invalid (cols. 49 through 53).
The program halts at address 02A6.
Feasibility condition: NO.

2516 x OUTPUT TAPE BLOCK NUMBER ERROR

The code indicating the number of blocks to be retained on the tape to be written is invalid (cols. 54 through 58).
The program halts at address 02A6. Feasibility condition: NO.

2517 x UNIT N : NON-STANDARD FILE

The unit specified has a non-standard file. The program halts at address 02A6.
Feasibility condition: NO.

2518 x END OF REEL

End of the tape being written.
When this message is printed out, one or the other of the two following conditions exists:

a) duplication has been completed: in this case, the program comes to a standard halt.
Feasibility condition: YES.

b) duplication has not been completed: in this case, the program halts at address 02A6 and the feasibility condition is set to NO.

The tape being written is closed in the following manner:

- if the tape contains programs, it is closed with a Tape Mark and a label composed of 28 exclamation mark characters(!).
- if the tape contains data with a standard format, it is closed with a tape mark and an EOF label in which the block count field is incorrect.
- if the tape contains date with a non-standard format, it is closed with a Tape Mark.

2599 x END OF TAPE DUPLICATE

End of the program. The order has been executed correctly.

6. EXAMPLE

A file containing standard-format data (written with odd parity check and with a recording density of 800 bits per inch on a volume with the serial number 000015, and mounted on a tape unit having the logical number 2) is to be duplicated on another tape with the same type of check and with the same recording density, having the serial number 000024 and mounted on the tape unit with the logical number 3.

The whole of the tape is to be duplicated and nothing is to be added; the serial number and the name of the file are not to be changed; the purge date is to be the day's date; all the blocks on the tape to be read should be duplicated but none of those which may be found on the tape to be written should be retained.

The following order card will be punched:

Col. 1 to 4	⊙TDC
Col. 5 to 6	DS
Col. 7 to 8	b̸2
Col. 9	P
Col. 10	L
Col. 11 to 16	000015
Col. 17 and 18	b̸3
Col. 19	P
Col. 20	L
Col. 21 to 26	000024
Col. 27 to 80	Not used.

•DFG DIAGNOSTIC FILE GENERATE

1. FUNCTIONS

The purpose of this program is to write on cylinders 98 - 99 of any volume a data file to be used by the diagnostic programs.

2. SYSTEM UNITS REQUIRED

The program requires the use of the following peripherals:

- a disc unit on which an initialized disc has been mounted on the basic system disc
- a card reader
- a printer
- another disc unit when the diagnostic file disc is not the basic disc.

3. PARAMETER CARDS

This program requires only one order card with the following format:

Col. 1 to 4	Identification code: •DFG
Col. 5 to 6	May contain one of the following values: - SD, if the diagnostic file disc is the System Disc - WD, if the diagnostic file disc is the Work Disc, - b̸n, where n is the logical number (from 0 - 9) of the diagnostic file disc.

Col.	7 to 72	Not used
Col.	73 to 80	Serial code if required.

4. DESCRIPTION AND EXECUTION

The program consists of a single segment at address 2048 (decimal).

The program first checks that the diagnostic file label has been written on the receiver disc, specified by the order •FLW.

All the headers of cylinder 0 are then checked against the contents of the memory.

The following headers are also checked:

00200
00600
00700
00800
00900
01000
01400
02400
04400
07000
08400
10000
10200
10400
16400

If this check is not complete, the generation of the diagnostic data file is interrupted.
When the receiver disc is not the System Disc, the order •DFG is executed. This implies that the order •IND has already been executed. This order writes the headers and duplicates cylinder 0.

The program then uses cylinders 98 and 99 to write specific character configurations to be used by the diagnostic program.

5. CHECKS AND ERROR MESSAGES

The program checks the disc unit involved and the validity of the order card. Any error causes one of the following messages to be printed:

4101 x D. UNIT N PERSISTENT ERROR

A persistent read, write or check error has been found on the unit with logical number N.
The program stops.
Feasibility condition: NO.

4102 x D. UNIT N OUT OF ORDER

Disc unit with logical number N has been found out of order.
The program slews the paper on printer to the next page, and comes to a halt at address 01DA.
Pressing RUN retries the operation that has caused the message.
If the message is repeated, the program stops.
Feasibility condition: NO.

4103 x D. UNIT N PERSISTENT ERROR

A persistent positioning error has been encountered on disc unit with logical number N.
The program stops.
Feasibility condition: NO.

4104 x OUTPUT D. UNIT NUMBER ERROR

The number of the diagnostic file disc unit as specified in the order card is incorrected.
The program stops.
Feasibility condition· NO.

4105 x TRACK XXXXX NOT AVAILABLE

One of the checks denotes lack of headers listed above.
The program stops.
Feasibility condition: NO.

4106 x DIAGNOSTIC FILE NOT FOUND

Check denotes lack of diagnostic file lable.
The program stops. Feasibility condition: NO.

4199 x END OF DIAGNOSTIC FILE GENERATE

Denotes the end of diagnostic file generation.
Control is transferred to the Operating System Pilot.

6. EXAMPLE

The diagnostic data file is to be written on a volume defined as Work Disc in col. 17 of order •INT.

The order card must be punched in the following way:

Col. 1 to 4	•DFG
Col. 5 to 6	WD
Col. 7 to 80	Not used.

•VLC VOLUME LABEL CHANGE

1. FUNCTION

To change the volume label and/or the volume name on a disc pack.

2. SYSTEM UNITS REQUIRED:

- A disc unit for the System Disc
- A printer
- Additional disc if its label is to be changed.

3. PARAMETER CARDS

The format of the order card is:

Col. 1 to 4	•VLC
Col. 5	Logical unit number of the disc to be changed
Col. 6 to 11	New volume label
Col. 12 to 21	New volume name
Col. 22 to 72	Not used
Col. 73 to 76	Serial code (optional)
Col. 77 to 80	Not used

4. DESCRIPTION AND EXECUTION

This program is made up of a single segment stored at memory location 1800 (decimal).

The program performs the requested changes and prints out the following messages

```
LOGICAL UNIT NUMBER N
OLD VOLUME LABEL XXXXXX OLD VOLUME NAME XXXXXXXXXX
NEW VOLUME LABEL YYYYYY NEW VOLUME NAME YYYYYYYYYY
```

Each file label is updated to reflect the change if the volume order number is equal to 0001 (positions 28 to 31 of HDR label).

5. CHECKS AND ERROR MESSAGES

6901 * D. UNIT X POSITIONING ERROR
6902 * D. UNIT X OUT OF ORDER
6903 * D. UNIT X PERSISTENT ERROR

Program execution terminates. Feasibility condition: NO.

6910 * ORDER CARD COLUMN 5 ERROR

The logical unit number is not between 0 and 9.
Program execution terminates. Feasibility condition: NO.

6950 * WARNING - FILE TABLE NOT UPDATED

Following a disc error, signalled by messages 6901-6902-6903, the file labels have not been modified. Program execution terminates; feasibility condition: NO.

6998 * VOLUME LABEL CHANGE FAILED

This message is printed after the messages 6901 - 6902 - 6903 - 6910. Program execution terminates; feasibility condition: NO.

6999 * VOLUME LABEL ENDED

End of program.

APPENDICES

APPENDIX A. PROCESSING THE LBC CARDS

A.1. FILE OPENING

Before a file on tape or on disc is processed, it must be opened, that is it must be checked and set.

The checks have a double purpose:

In the case of an input file they guarantee its identification. If the file is being written, they prevent erasure of valid information. The checks can apply to the file name, the writing date, the cancellation date, the file serial number and the sequence of the volumes making up one file.
Some of these checks are automatic and some are plicitly required by the LBC card.

The opening of a file is carried out by the macro-instructions OPEN and NEXT described in the APS II Language Manual. The macro-instructions check the data stored in the file table against the information contained in the file label.

The file table is made up of all the parameters that are necessary to open and control the files concerned in a program. It built up by the compiler APS II which retrieves data from the FILE instructions and reserves the locations which are to contain the data supplied by the parameter cards (LBC cards).

The opening of a file implies the compiling of an LBC card that contains the data from which the file label checks must be performed and the storing of these data into the program file table. Such storing can be carried out at three different times:

a. when the program is written on the Library Disc.

b. when the program is written on the Master Disc.

c. when the program is initiated.

The following orders apply to the LBC cards:

•LDU, •MDG, •PRL and •LPC.

Each time one of these orders applies to an LBC card it transfers into the file table only the data that have been written on that card and leaves unchanged in the table the data not specified in the card.
In the same way the LBC card may be used to modify a single data in the file table.
The next paragraphs describe the file table relative to files on tape/disc the LBC card format and the card processing mode.

A.2. LBC CARD FORMAT

Each leader card contains the parameters relative to a given file. It has the following format:

Col. 1 to 4	card code, ƀLBC (Label Control)
Col. 5 to 12	program name relative to the file to be opened
Col. 13 to 18	file internal name used within the program in the FILE instruction and the processing instructions referring to the file
Col. 19 to 30	file external name written on label HDR
Col. 31 to 36	file serial number
Col. 37 to 42	ƀYYDDD representing the file generation date. This date may also be indicated indirectly with reference to the area reserved for the user by the Operating System. Depending on the values contained in column 42 (from 0 to 5) the generation data source is as follows:

0 - from locations 40 - 45
1 - from locations 52 - 57
2 - from locations 58 - 63
3 - from locations 64 - 69
4 - from locations 70 - 75
5 - from locations 76 - 81

Col. 43	number of the unit on which the volume containing the file or the first part of the file is mounted
Col. 44	number of the alternate unit used for sequential processing of a multivolume file. If the file is always read by the same unit the number indicated in column 43 is repeated. If the is constituted by a single volume column 43 is not punched
Col. 45	file external name check option; the option may be: Y - if check is desired N - if check is not desired In the case of files on disc, the check is automatic and column 45 is not used
Col. 46	file serial number check option: Y - if check is desired N - if check is not desired
Col. 47	volume sequence check option for the processing of a multivolume sequential file. The option is Y or N.
Col. 48	file generation date check option (Y or N)
Col. 49	processing option (Y or N) when file is to be processed as an absent file. This option is only valid with input files. If it is Y, all the operations relative to the file are cancelled.

Col. 50	print out option expressed as the preceding one. If the option is Y, the program which opens the file prints out the HDR label. When the check detects an error, the program prints out a part of the file table.
Col. 51	file packing density. It can assume one of the following values: L = 800 bpi M = 556 bpi N = 200 bpi Column 51 is not used for file on disc.
Col. 52 to 80	not used.

The dates contained in the user's area are memorized by means of the orders •INT, •PRL, •MDG.

The file serial number and the generation date are indicated only when it is desired to make sure that they match with the information written on HDR label, that is when the relative options (col. 46 and 48) are specified as Y.

The parameters found in columns 43 - 44 are also indicated in the instruction FILE. They are optional in the leader card by when they are present they take precedence over the parameters contained in the instruction FILE.

A.3. DESCRIPTION OF THE TABLE FOR FILES ON TAPE

For each file on tape the following characters are reserved in the file table:

Col. 1	code indicating the type of the file; it can assume the values: 00 when the file is constituted by FL or UN records 01 when the file is constituted by VL records

Col. 2	number of characters reserved for the file in the file table: 94 (decimal) or 5E (hexadecimal).
Col. 3 to 6	HDR1
Col. 7 to 23	reserved for the file external name read on LBC card
Col. 24 to 29	reserved for the file serial number read on LBC card
Col. 30 to 33	reserved for the volume sequence number in the file
Col. 34 to 37	File sequence number: 0001
Col. 38 to 43	reserved for the first volume serial number in the file
Col. 44 to 49	reserved for the file generation date read on LBC card
Col. 50 to 55	file retention period (parameter RP) contained in the FILE instruction or decimal zeros when parameter RP is not present. The cancellation date is calculated in that area when the program is initiated.
Col. 56	working location
Col. 57 to 62	file internal name taken from FILE instruction. After the opening of the file these locations are used to count the blocks.
Col. 63 to 68	check options corresponding to the columns 45 - 50 of the LBC card. The compiler previously supposes options YYYYNY.
Col. 69	Number of the alternate unit.
Col. 70 to 85	Reserved for the parameters of the tape subroutine.
Col. 86	Code indicating the description of the file: 00 = standard file with numbered blocks 01 = standard file with non numbered blocks 80 = non standard file with numbered blocks 81 = non standard file with non numbered blocks

Col. 87 to 88	length of the record area (WA)
Col. 89 to 90	address of the record area
Col. 91 to 94	working storage.

A.4. DESCRIPTION OF THE FILE TABLE FOR FILES ON DISC

The description of the file table for files on disc remains valid only until the macro-instruction OPEN is executed.

The following characters are reserved in the file table for each file on disc:

Col. 1	code indicating the type of file processing: 10 = sequential processing with FL or UN records 11 = sequential processing with VL records 20 = random processing 30 = mixed processing with FL or UN records 31 = mixed processing with VL records
Col. 2	length of the parameters relative to the file. This length equals 100 characters for sequential files and all monovolume files. For random or mixed multivolume files this length is calculated as follows: 100 + 10 n with n standing for the number of volumes making up the file
Col. 3 to 4	end of file/volume condition check subroutine address (instruction file parameter EV)
Col. 5 to 6	FA area address. Binary zeros if FA has not been specified
Col. 7 to 18	reserved for the file external name read on the LBC card

Col. 19 to 20	DA area address. Binary Zeros if DA has not been specified.
Col. 21	KA area length. Binary zero if KA has not been specified.
Col. 22 to 23	KA area address. Binary zeros if KA has not been specified.
Col. 24 to 29	Reserved for the file serial number read on the LBC card.
Col. 30 to 33	Reserved for the volume sequence number in the file.
Col. 34 to 35	RA area address. Binary zeros if RA has not been specified.
Col. 36 to 37	Page length decimal expressed in number of sectors.
Col. 38 to 43	Working locations.
Col. 44 to 49	Reserved for the file generation date read on the LBC card.
Col. 50 to 55	File retention period (parameter RP in the File instruction). Decimal zeros if RP has not been specified. When the program is initiated the retention date will be calculated inside these locations.
Col. 56	Reserved for a code that can assume one of the following values: 1 = file maintenance 0 = no file maintenance
Col. 57 to 62	File internal name taken out from the FILE instruction
Col. 63	Work location
Col. 64 to 68	Check options corresponding to the columns 46 - 50 of the LBC card. The compiler APS supposes options YYYNY when generating the file table.
Col. 69	Alternate unit number, or value of parameter FE if processing is random or mixed.

Col. 70 to 85	reserved for the disc subroutine parameters
Col. 86	code indicating the type of file: S = sequential file R = random file
Col. 87 to 88	WA area length (binary). Zero if WA has not been specified.
Col. 89 to 90	WA area address (binary). Zero if WA has not been specified.
Col. 91 to 100	Work area.

With random/mixt multivolume files ten supplementary locations containing the following information expressed in packed characters are reserved for each volume:

5 characters - overflow area first sector address
5 characters - overflow area first external sector address
5 characters - overflow area first free sector address
5 characters - file label address.

A.5. PROCESSING OF LBC CARDS

All the orders that apply to LBC cards carry out the following functions:

1. Check the program name against the name in columns 5 - 12
2. Search in the file table, for the element relative to the file whose name is indicated in columns 13 - 18
3. Store into the file table the file external name found in columns 19 - 30
4. Store into the file table the file serial number found in columns 31 - 36.

5. Perform a formal check on the date written in columns 37 - 42. This date is expressed in one of the following forms:

   a. 1 character b̸ and 5 numeric characters
   b. 5 characters b̸ and 1 numeric character ranging from 0 to 5.

6. Store the date in the file table. If the table is expressed as under 5.b the orders •LDU and •MDG store the code instead of the actual date. The actual date will be taken from the user's area and stored in the file table when the order •PRL (or Monitor) is issued.

7. Calculate the cancellation date by means of the order •PRL (or Monitor).

8. Perform a formal check on the number specifying the primary unit. This number must range from 0 to 9 in the case of a file on disc and from 1 to 8 in the case of a file on tape.

9. Store this number into the file table.

10. Perform a formal check on the number specifying the alternate unit. That number must be expressed as the preceding one. When the file is written on disc the check is performed only if processing is sequential.

11. Store this number in the file table.

12. Perform a formal check on the options expressed in columns 45 - 50. Column 45 is checked only if the file is written on tape.

13. Store those options in the file table.

14. Perform a formal check on the code indicating the packing density (col. 51). This code must have the following values: L M N .

15. Store this code into the file table.

It should be noted that only the data actually written in the LBC card are stored in the file table. When a data is not found on the card, the corresponding data in the file table will not be modified.

APPENDIX B. PILOT ZERO CARD FORMAT

The Pilot 0 initialization program (see. sub-paragraph 4.2.1., page 27) is punched in pure hexadecimal on one card. Its format is as follows:

a) For systems using Hollerith punch code:

Col. 1 and 2	97
Col. 3 and 4	Connector number (in hexadecimal) of the disc unit on which the system's basic program file is mounted (x)
Col. 5 to 10	00229>
Col. 11 to 12	(same as cols. 3- 4)
Col. 13 to 18	00209>
Col. 19 to 20	(same as cols. 3 - 4)
Col. 21 to 26	001[9>
Col. 27 and 28	(same as cols. 3 - 4)
Col. 29 to 80	001@4320010043?1000000021040@04401005046504[001500 19

b) For systems using the Bull punch code:

Col. 1 and 2	ZU
Col. 3 and 4	Connector number (in hexadecimal) of the disc unit on which the system's basic program file is mounted (x)

(x) Following is a correspondence table for coded and pure hexadecimal values.

Col. 5 to 10	b̸b̸55ZU
Col. 11 and 12	(same as cols. 3 - 4)
Col. 13 to 18	b̸b̸5b̸ZU
Col. 19 and 20	(same as cols. 3 - 4)
Col. 21 to 26	b̸b̸6YZU
Col. 27 and 28	(same as cols. 3 - 4)
Col. 29 to 80	b̸b̸6W345b̸b̸6b̸b̸34T6b̸b̸b̸b̸b̸b̸56b̸3b̸Wb̸33b̸6b̸b̸2b̸312b̸3Yb̸b̸62b̸b̸6Z

CORRESPONDING PURE HEXADECIMAL

CODE HEXADECIMAL	CODE HOLLERITH	CODE BULL
0	0	b̸
1	1	6
2	2	5
3	3	4
4	4	3
5	5	2
6	6	1
7	7	0
8	8	9
9	9	Z (9 - 6)
A	[(8 - 2)	Y (9 - 5)
B	# (8 - 3)	X (9 - 4)
C	@ (8 - 4)	W (9 - 3)
D	: (8 - 5)	V (9 - 2)
E	> (8 - 6)	U (9 - 1)
F	? (8 - 7)	T (9 - 0)

Example: if the connection number of the disc unit containing the System Disc is 07 (coded hexadecimal) the punch will be 07 in the specified columns (using code Hollerith) or b̸0 (using code Bull).

DOCUMENT REVIEW SHEET

TITLE: GE-100 Line DOS Disc Operating System
CODE: 3916163 F / Réf. 32.38.003 A

FROM:
Name :
Position:
Address:

Comments concerning this publication are solicited for use in improving future editions. Please provide any recommended additions, deletions, corrections, or other information you deem necessary for improving this manual. The following space is provided for your comments.

COMMENTS:

Please cut along this line

Fold on two lines shown on reverse
side, staple, and mail.

FOLD

Postage

HONEYWELL
INFORMATION SYSTEMS ITALIA
MARKETING SUPPORT
Documentation
and Software Distribution
via Pirelli, 32
20124 MILANO
(ITALIA)

FOLD

DOCUMENT REVIEW SHEET

TITLE: GE-100 Line DOS Disc Operating System
CODE: 3916163 F / Réf. 32.38.003 A

FROM:
Name :
Position:
Address:

Comments concerning this publication are solicited for use in improving future editions. Please provide any recommended additions, deletions, corrections, or other information you deem necessary for improving this manual. The following space is provided for your comments.

COMMENTS:

Fold on two lines shown on reverse side, staple, and mail.

Please cut along this line

FOLD

Postage

HONEYWELL
INFORMATION SYSTEMS ITALIA
MARKETING SUPPORT
Documentation
and Software Distribution
via Pirelli, 32
20124 MILANO
(ITALIA)

FOLD